JOINT SOURCE-CHANNEL CODING WITH REAL NUMBER BCH AND REED-SOLOMON CODES: THEIR PROPERTIES AND PERFORMANCE IN THE PRESENCE OF ADDITIVE NOISE

By

JOHN DAVID ENDSLEY

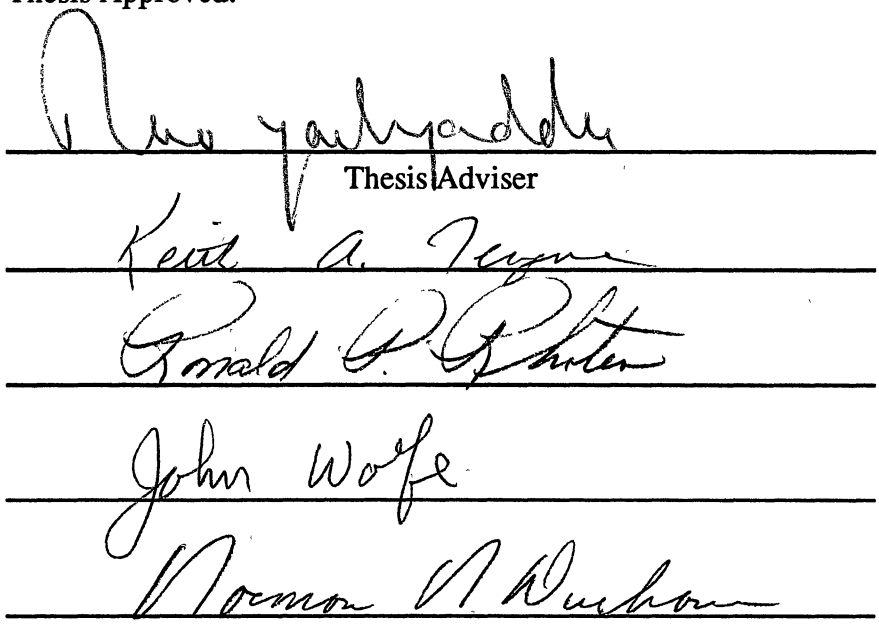
Bachelor of Science in Electrical Engineering
New Mexico State University
Las Cruces, New Mexico
1985

Master of Science
University of Colorado, Boulder
Boulder, Colorado
1988

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 1991

JOINT SOURCE-CHANNEL CODING WITH REAL NUMBER BCH AND REED-SOLOMON CODES: THEIR PROPERTIES AND PERFORMANCE IN THE PRESENCE OF ADDITIVE NOISE

Thesis Approved:



Thesis Adviser

Dean of the Graduate College

PREFACE

This thesis investigates the joint source-channel coding properties of real number BCH and Reed-Solomon codes in the presence of additive noise. From previous results, it was known that additive noise can cause the error correction ability of a real number code to degrade. This degradation results in decoding failures. Knowing this, there are two main objectives of this research. The first objective is to determine under what conditions a given real number code is reliable. More specifically, for a given real number BCH or Reed-Solomon code, I sought to determine the highest additive noise level for which the real number code could still be accurately decoded within a specified probability of failure. Using these results, the second objective is to determine whether a real number code can obtain better joint source-channel performance than a comparable finite field code.

During the investigation process, I formalized the source coding properties that had been mentioned in previous research. The first objective was met by deriving an upper bound to the probability of a decoding failure as a function of the signal to noise ratio, the transmission error magnitudes and the code parameters. These bounds assume that a full search decoding method is implemented.

Since the full search method is impractical and the traditional decoding method performed poorly in the presence of additive noise, an alternate decoding algorithm was developed. This algorithm attempts to combine the directness of the traditional BCH decoding algorithm with the robustness of the full search decoder.

The second objective was met with mixed success since deriving an accurate average channel coding performance for multiple error correcting codes proved elusive. However, simulated results for a four error correcting code is examined.

This research could not have been accomplished without the support of the Department of Defense under contract number MDA 904-88-6017 and Sandia National Laboratories under contract number 63-0906. I wish to thank my major adviser, Dr. Rao Yarlagadda for his guidance and assistance during the course of this research. In addition, I want to extend thanks to my other thesis committee members, Dr. Ron Rhoten, Dr. Keith Teague, Dr. John Wolfe and Dr. Roger Zierau for their support.

Most of all, the I would like to thank my parents David and Joan Endsley for their love and constant encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

Symbol	Explanation	Page[1]
$\mathbf{R}$	Field of Real Numbers	34
$\mathbf{R}^N$	Real Vector Space Consisting of all $N \times 1$ Vectors	160
$\mathbf{C}$	Field of Complex Numbers	34
$\mathbf{F}_q$	Finite Field with q Elements	36
d_{min}	Minimum Distance For a Given Code	41, **45**
d	Length K Information Word	42
c	Length N Codeword	42
G	$N \times K$ Generator Matrix: $c = Gd$	42
G_{L^c}	Deleted Generator Matrix: $G_{L^c} = S_{L^c}^T G$	47, 73
G^+	Pseudo-inverse of G	**167**
H	$N \times N-K$ Parity Check Matrix: $H^T G = 0$	42
C	Codespace	42, **42**
$C^\angle$	Orthogonal Complement to C	43
e	Transmission Error Vector	43
r	Received Data Word: $r = c + q + e$	43
s	Syndrome Vector: $s = H^T r$	44
L, J	Index Set: $L, J \subset [0, \ldots N-1]$	44
$\lvert L \rvert$	Number of Indices in L	44

1 Bold face indicates that the symbol is defined on this page.

Symbol	Explanation	Page
H_L	Deleted Parity Check Matrix	44
t	Maximum Number of Transmission Errors a Code Can Correct	45, **46**
L^c	Complement of L w.r.t. $[0,\ldots,N-1]$	**74**
μ	Number of Transmission Errors	44
$\lfloor . \rfloor$	Floor Function: Keeps the integer part.	46
W_N	$N \times N$ Discrete Fourier Transform Matrix	53
$G_{\mathbf{R}}, G_{\mathbf{C}}$	Real and Complex Generator Matrices	57, **77**
$H_{\mathbf{R}}, H_{\mathbf{C}}$	Real and Complex Parity Check Matrices	57, **77**
$\hat{d}$	Estimate of the Information Word	67
MSE	Mean Squared Error for d and $\hat{d}$	67
$Q[\cdot]$	Quantization Operation	68
q	Quantization Error Vector	68
$\mathrm{MSE}_{\mathrm{FF}}$	Total MSE in Finite Field Case	68
$\mathrm{M1}_{\mathrm{FF}}$	MSE Given $M \leq t$ in Finite Field Case	68
$\mathrm{M2}_{\mathrm{FF}}$	MSE Given $\mu > t$ in Finite Field Case	68
P_{E}	Probability of Channel Decoding Error	68
$\mathrm{MSE}_{\mathrm{RN}}$	Total MSE in Real Number Case	69
$\mathrm{M1}_{\mathrm{RN}}$	MSE Given $\mu \leq t$ in Real Number Case	69
$\mathrm{M2}_{\mathrm{RN}}$	MSE Given $\mu > t$ in Real Number Case	69
$\mathrm{P}_{\mathrm{UNC}}$	Probability of an Uncorrectable Transmission Error Vector	69
P_{DF}	Probability of a Decoding Failure	69
S_L	Selection Matrix: Selects Rows Specified By L	**78**

Symbol	Explanation	Page
R_d	Correlation Matrix for d	80
σ_q^2	Quantization Noise Variance	80
$\tilde{G}$	Weighted Generator Matrix	**85**
Δs	Syndrome Perturbation	87
$\overline{s}$	Syndrome Mean	87
P_J	Orthogonal Projection onto $Im(H_J^T)$	88, **166**
$\theta_{min}(L)$	Minimum Angle Between $Im(H_L^T)$ and Closest Error Syndrome Subspace	92, **100**
P_C	Probability of Decoding Correctly	93
$erf(\cdot)$	Error Function For Gaussian Density	**93**
P_S	Probability of a Symbol Error	100
$\theta_{avg}(\mu)$	Average Minimum Angle as a Function of the Number of Errors	99, **100**
$\xi(\mu)$	ANSD Threshold as a Function of the Number of Errors	120, 125

CHAPTER I

INTRODUCTION

The reliable transmission of real or complex valued data is a common requirement in many of today's communication systems. Consequently, many coding methods have been devised. Ideally, each of these methods strives to encode the real or complex numbers so that after transmission, the received data perfectly matches the original values; i.e. no errors (also called noise or distortion) have been introduced by the transmission process.

Of course, it is well known that it is impossible to transmit a real or complex number and receive the value precisely. This is due to the fact that perfect reception of a real or complex number requires an infinite capacity channel. Such a channel is not physically realizable. The best that can be hoped for is to receive some sort of accurate facsimile of the original data.

The distortion in the received data can be divided into two categories:

1. Distortion introduced by reducing the real or complex valued data into a finite representation. (Only digital communication systems are considered in this document.)
2. Distortion introduced by random or impulsive transmission noise in the channel.

It is customary to reduce these two causes of distortion independently, by using specialized encoding procedures. Loosely speaking, a *source coder* tries to minimize the first cause of distortion, while a *channel coder*, attempts to minimize the second.

This document contains the results of the author's investigation into one possible method for encoding real and complex valued data. This method combines the two usually separate processes of source and channel coding; thus, this method is categorized as a *joint source-channel* coding procedure.

Many traditional channel coding methods commonly make use of certain algebraic error correcting codes called Bose-Chauduri-Hocuenghem (BCH) and Reed-Solomon (RS) codes. These codes usually encode data which is derived from a finite alphabet. For example, a binary code would encode binary data. Certain RS codes might encode bytes; a byte, being a group of eight bits, would represent an element from a finite alphabet with 64 different symbols. These codes are designed to correct transmission errors, which in turn, ensures that the communication channel is reliable.

An algebraic structure can be imposed on the alphabet which results in a *finite field.* Informally, a field is a collection of elements in which any two members can be either added, subtracted, multiplied or divided. If this collection has a finite number of members, then it is a finite field. Similarly, an infinite field has an infinite number of elements. For example, the fields of real and complex numbers are both infinite fields.

Traditional BCH and RS codes are based upon finite fields. This document investigates non-traditional BCH and RS codes which are based upon the real and complex fields. Because the analysis of finite and infinite fields varies significantly, real number BCH and RS codes have different properties than their finite field counterparts. (Note: Henceforth, the term real number (RN) codes will collectively refer to codes based on either the real or complex fields.) The source and channel properties of real number BCH and Reed-Solomon codes, along with their performance in the presence of additive noise, are investigated in this document.

Motivation

As mentioned previously, the transmission (or storage) of real or complex valued numbers (represented by two real numbers) is a common requirement in modern communication and computer systems. In fact, with the growing availability of extended precision Digital Signal Processors (DSPs) and inexpensive high-performance computers, the demand to transmit or store real valued data will only increase.

Of course, a number contained in a storage register within a processor is represented by a finite number of bits. So strictly speaking, this number is a member of a finite alphabet. However, this value is a representation of a real number, and is equal to the real number plus an error term. Hopefully, if the numerical manipulations which led to the real valued result are well conditioned and sufficient precision is available to the processor, then this error term will be small.

Errors caused by the finite number of register bits available to a processor for representing a real number, will be called *roundoff noise*. For the most part, this discussion is not concerned with roundoff noise. Instead, more attention is focused on what is called *quantization noise*. This type of error occurs in the source coding process when the machine representation of a real number must be reduced for storage or transmission.

The difference here is subtle and the notation is not standard. For example, a system that processes 32 bits and also transmits (or stores) the full 32 bit representation is susceptible to roundoff noise, but not quantization noise. On the other hand, a 64 bit representation might be used in processing while 32 or 16 bit representations are used for storage. In this latter case, assuming the processing operations are well conditioned, the quantization noise is dominant and should be the major concern.

As a final example, consider a system that samples a portion of an analog waveform using a 16 bit Analog-to-Digital Converter (ADC). A 32 bit DSP is then used to process the data, yielding a vector of 32 bit numbers. For transmission, these numbers are then

reduced to 8 bits. The first and third steps both introduce quantization noise, while the second step suffers from roundoff noise. However, assuming the system is properly designed, the quantization noise of the third step dominates the other two sources of error. Subsequently, it can be approximated as the only source of error, with perfect real number representations preceding it.

Both speech and image processing systems are similar to this last example. In addition, it is common that the output of these systems must be transmitted over channels that are susceptible to transmission errors. A coding procedure for these systems attempts to minimize the quantization errors, while also providing channel error protection. In order to illustrate when a real number error correction code would be used, consider the following hypothetical example.

Suppose that a processor has a block of real numbers that are to be transmitted to a user. Furthermore, suppose that because errors can occur during the transportation process, an error correction code is to be implemented as part of the coding process. It is assumed that, with the exception of the final quantization, the source coding process is complete.

With a traditional error correcting code, the numbers will be quantized to a fixed number of bits, thus fixing the average level of the quantization noise. Next, a finite field based error correction code will be applied to the quantized data. Inherent in the application of the error correction code is the addition of overhead or redundant bits, which are used for detecting and correcting errors. For this case, the source and channel coding processes are independent.

A non-traditional approach using a real number error correction code, would first apply the RN code, and then quantize the coded data. In this case, the two processes are not independent.

In the subsequent chapters of this document, it will be shown that the added redundancy of the RN error correcting code can reduce the quantization noise to a lower level than what would be obtained by the traditional finite field approach. However, as a drawback, the quantization noise affects the error correction procedure. With the RN code, there is a finite probability that the error correction code will fail, even though the number of errors has not exceeded the designed error correction capabilities of the code.

It will also be shown that the probability of a decoding failure is a function of the parameters of the code, the magnitude and direction of the transmission error, and the level of the quantization noise.

Overview of Document

This document is comprised of eight chapters. In addition, two appendices are included. The purpose to this first introductory chapter is to present some general motivational framework for real number codes, an overview of the entire document, a discussion of the previous work in the area of real number error correcting codes, and a discussion of the contributions of the author's research. A limited knowledge of real number codes is presumed for these final two sections.

Chapter II contains an elementary overview of a typical digital communication system. The basic definitions and ideas behind an information source, source coding, channel coding, and an information channel are presented. The reader familiar with these concepts and terminology may want to only skim this chapter.

Chapter III presents the basic ideas and terminology of error correction codes. It begins with a brief discussion of fields; examples of infinite fields along with the more abstract finite fields are given.

In addition, a matrix description of linear block codes is presented. The basic ideas behind single error correcting and multiple error correcting codes are discussed. Similar to Chapter II, the reader familiar with traditional error correcting codes may want to only browse this chapter.

Chapter IV is devoted to real number BCH and Reed-Solomon codes. The definition for these codes is based upon the complex Discrete Fourier Transform (DFT) matrix. Both codes are defined, and certain normalizing properties are presented. The Prony algorithm is derived for decoding these codes.

Chapter V discusses the properties of these real number codes in the presence of additive quantization noise. This discussion is divided into three parts: the source coding aspects, the channel coding aspects, and the combined source-channel coding aspects. This chapter contains the majority of the author's theoretical results concerning RN BCH and RS codes.

Chapter VI contains the main decoding methods for single and multiple error correcting codes in the presence of quantization noise. It begins with a discussion of some of the previously proposed methods for decoding RN codes. The advantages and disadvantages of these methods are presented. It concludes with the derivation of another decoding method which combines some of the positive aspects of the previous methods.

Chapter VII contains the results of the simulations which attempt to verify the theoretical results of chapters V and VI. It includes source coding simulations, channel coding simulations, and finally joint source-channel coding simulations. Using the simulation results, Chapter VII also discusses the realistic properties of RN codes as compared to the theoretical properties derived earlier.

Chapter VIII concludes by summarizing the major contributions of the research contained in this document. In addition, this final chapter also comments on the application of real number codes to real communication systems.

The two appendices at the end of this document contain a brief review of matrix algebra, and a discussion of the singular value decomposition. The reader unfamiliar with these topics may want to review these appendices before Chapter IV.

Previous Work

Although the discipline of finite field error correcting codes is fairly mature, with some of the original ideas dating back to the late 1940's and early 1950's, ([Sha48], [Ham50]), the study of real number codes is relatively recent. The majority of work in real number error correction codes has been performed in the middle 1980's, which is a bit surprising since real number codes are much more accessible to the average engineer than their finite field counterparts. Real number codes require no knowledge of abstract algebra, and the popular BCH and Reed-Solomon versions of these codes can be succinctly described by the familiar Discrete Fourier Transform.

One of the first links between finite field codes and real number analysis techniques was made by Wolf in 1967, [Wol67]. He realized that decoding finite field BCH codes is conceptually identical to Prony's method of exponential curve fitting. This method was discovered by Prony in the late 18th century, [Pro95]. The method is common in the sinusoid estimation literature, [Kay88], [Mpl87]. Although Wolf did not propose or define real number codes at the time, he would later refer back to this original work in his later investigations.

It appears that Marshall was the first to actively pursue real number codes, [Mar81]. This paper, titled, "Real Number Transform and Convolutional Codes", laid many of the basic foundations for real number codes. Marshall recognized that, first, real number codes could have some advantages over their finite field counterparts in that there now exists an expanding supply of signal processing hardware that could be easily programmed to manipulate the real numbers. Decoding of finite field codes generally

demanded some sort of specialized hardware. Today, there exist integrated circuits that can implement many of the popular finite field codes; however, in the past, these codes usually demanded custom hardware which could be expensive and tedious [Ber68], [Chi64], [Pet60].

The second advantage that Marshall recognized was the possibility of combined source-channel coding with real number codes. The principles that lead to Marshall's "bandwidth compression", are essentially the same ideas that lead to a reduction in the overall quantization noise level discussed in this document. For a fixed quantization noise level, (distortion level), a real number code can be used to reduce the bandwidth; or alternately, for a fixed transmission rate, the RN code can obtain a lower distortion level.

No results enumerating how much bandwidth can be saved were shown by Marshall. In addition, no work discussing the reliability of real number codes in the presence of quantization noise was presented. In his original paper, Marshall also proposed that real number codes can be used in a concatenated coding scheme with lower level finite field codes.

Whereas some finite field linear block codes are commonly analyzed by using polynomial techniques, (especially cyclic codes), real number linear block codes are best suited for analysis with transform techniques. The transform viewpoint of error correction codes was encouraged by Blahut, [Bla77]. He felt that the transform viewpoint made error correction codes more accessible to the general electrical engineering community. Blahut, like Wolf before him, did not propose real number codes at the time. However, he too would later define and discuss BCH and RS codes over the real and complex fields.

After Marshall's work, Wolf published a correspondence paper entitled, "Redundancy, the Discrete Fourier Transform, and Impulse Noise Cancellation", [Wol83b]. In this paper, Wolf recognizes the possible value of real number error correction codes. He also examines the relationship between these codes and the DFT.

One of the main results of this paper is that Wolf explains why a real number DFT code is capable of correcting up to twice as many errors as would be possible over a finite field code. The method will be called the "voting argument" and will be discussed further in Chapter VI; for right now, it will only be noted that the voting argument is not a practical decoding algorithm for larger codes.

Wolf concludes his paper with a single error decoding example using the normal BCH decoding algorithm, (the one that is identical to Prony's method). This appears to be the first published numerical example. In his conclusion, Wolf states that the determination of how large the quantization noise can become before the decoding procedure breaks down is a subject for future study.

During 1982 to 1983, Marshall and his colleagues published several more articles on real number codes, [Mar82], [Mar83a], [Mar83b], & [Spr82]. Meanwhile, Wolf published a conference paper which reiterates his earlier ideas, [Wol83a]. In his papers, Marshall concentrates more on drawing parallels with the theory of digital filtering. He also discusses the technique of error trapping for decoding double error correcting codes. In all these discussions, it is mentioned that real number codes will suffer from the effects of roundoff or quantization noise; however, no analysis concerning these effects is presented. It is only mentioned that a "zero threshold" will have to be determined. The threshold will depend upon the noise level and it indicates when a number is, for all practical purposes, a zero. A number with magnitude smaller than the zero threshold is non-zero only because of the accumulation of roundoff or quantization noise.

In 1984, Marshall combined his previous ideas into a paper titled, "Coding of Real-Number Sequences for Error Correction: A Digital Signal Processing Problem.", [Mar84]. Here he reiterates and formalizes the existence and some of the properties of real number codes. The quantization noise problem is not addressed.

Until 1985, the areas of error correction and spectral estimation had not been connected except for the early paper by Wolf linking Prony's method to the BCH decoding algorithm. Prony's method is a means for approximating exponentials, [Hil50]. In the spectral estimation literature, many estimation techniques find their basis in Prony's method, since it provides a procedure for estimating the parameters of complex sinusoids in noise.

Modifications and extensions of the basic Prony method have resulted in new techniques that improve the estimation of exponentials in the presence of additive noise, [Kay88], [Mpl87], & [Tuf82]. The estimation literature provides analysis of these improved methods in the presence of noise, [Hua88], and also some newer matrix based methods, [Hua90]. However, for the error correction problem considered here, only the basic Prony method is discussed in detail.

In a 1985 *IEEE Proceedings* article titled, "Algebraic Fields, Signal Processing, and Error Control", Blahut ties much of the signal processing and coding viewpoints together. Much of the later work in real number error correction codes, including this document, is based upon this treatise. In "Algebraic Fields, ...", Blahut formally defines BCH codes and RS codes over the complex field. He shows how an extension field can be created using both finite and infinite fields. Conceptually, the construction of BCH and RS codes over finite and infinite fields is the same; however, since infinite fields are more familiar, BCH and RS codes defined over these fields are usually easier to grasp. Blahut also uses the real and complex fields to introduce the reader to BCH and RS codes in a recent digital communications book, [Bla90].

Three papers, published by Marshall, examine the real number error correction codes from the sequence interpolation and signal restoration viewpoints, [Mar85], [Mar86], & [Mar88]. In addition, several authors published results on real number error correction codes which draw upon some of the more recent spectral estimation techniques, [Kum85], [Sch87], & [Beh88].

The paper by Kumaresan performs one of the first simulations of real number codes that includes the effects of the quantization noise. In this paper, a (32,22) RS code is simulated for the case of four consecutive errors. He makes use of the rank reduction techniques developed in the estimation literature.

Two papers concerning real number error correction codes by Scharf and his colleagues, can also be found, [Sch87], [Beh88]. The second paper generalizes the ideas of the first, and it does not really address the error correction problem directly. The first paper simulates a (7,2) BCH code; it uses a matrix based statistical approach. This paper notes that in the decoding case where there are no errors, the extra redundancy present in the codeword can be used to reduce the "background" (quantization) noise. This paper uses a full search decoding technique, which is reminiscent of Wolf's voting argument in the sense that it is impractical for larger codes.

Very recently, real number codes have been examined for use with fault tolerant matrix operations, [Nai90]. This paper shows the existence of real number codes, and examines the use of these codes in the presence of roundoff noise. The goal for using these codes on processor arrays is mainly to detect the presence of incorrect matrix calculation results due to the failure of one of the array elements. These codes are mostly checksum codes and are simpler than the BCH and RS codes that are of interest in this document.

Contributions of This Research

As suggested by the title, this document examines the joint source-channel properties of real number BCH and Reed-Solomon codes. In addition, the reliability of these codes in the presence of additive noise is examined.

As Marshall and Scharf mentioned in their discussions, the extra redundancy introduced by the error correction code can be used to reduce the quantization noise level of the received data. This idea is formalized in this document and an expression is derived for the amount that the noise level can be reduced. It is found that the factor by which the noise level is reduced is a function of the code parameters, the number of errors, and also the error positions. Assuming that the different error positions are equally probable, an average noise reduction factor is calculated for a given number of errors.

These results, when combined with the characteristics of the channel, yield a final average joint source-channel distortion level. This is done for a memoryless symbol channel with a given number of quantization bits per codeword symbol. It must be assumed that the quantization noise level is sufficiently low so that the channel coding properties of the real number codes are reliable, i.e., the probability of a decoding failure is assumed to be negligible.

In addition to the above source coding results, weighted BCH and RS codes are derived. A weighted code allows the designer to specify that certain data symbols in a block of data are to be reproduced more precisely than other symbols. This is done through the use of a weighting matrix. It is shown that the decoding procedure is not affected by the weighting; only the source coding properties of the code are changed.

When analyzing the source coding characteristics of real number BCH and RS codes, it is assumed that the decoding is perfect; i.e., that the error positions can always

be correctly determined. However, in the presence of quantization noise this is not always so. Determining how robust a RN code is to additive noise is an important question.

In an early paper by Wolf, he left this question as an area of future research. Blahut, made a similar remark when referring to real number error correction codes:

"An important consideration is that of roundoff error. There is no roundoff error in a finite field. Each component of the received signal is either in error or not in error. It is quite specific to say how many components are in error. In the real field or in the complex field, however, there may be some minor errors in each component of the received signal, perhaps due to roundoff."

(He continues),

"However, to date there has been no theoretical work quantifying how big the minor errors can be before the error correction algorithms break down.", [Bla85].

In this document, results specifying the probability of a decoding failure are derived as a function of the signal to noise ratio (SNR) and the magnitude of the transmission errors. In general, the probability of a decoding error depends upon the parameters of the code, the level of the quantization noise, the transmission error locations, and the magnitudes of these errors. Worst case performance results are calculated and compared against simulated results.

A general decoding strategy is developed for a given, known quantization noise level. This strategy makes no assumptions on the nature of the channel. It decodes by assuming the maximum number of errors, and then systematically reduces this number by eliminating the errors whose magnitudes are small. The small errors are eliminated by comparing the magnitude to a threshold which is derived by minimizing a cost function based upon the expected mean squared error (MSE) of the data estimate. Some simu-

lations of real number codes using this algorithm are presented. Before proceeding with the properties of real number error correction codes, a brief introduction of the basic ideas of a digital communication system is needed.

CHAPTER II

DIGITAL COMMUNICATIONS OVERVIEW

The theoretical study of digital communications requires the discipline of information theory along with a good deal of mathematical sophistication. This document is concerned only with one particular coding technique, more from a signal processing viewpoint than from an information theoretic viewpoint. For this reason, this chapter only presents some of the basic ideas and terminology of digital communications. The interested reader can consult [Bla87], [Gal68], [Bla90], and [Skl88] for more detail.

Communication can be regarded as the transfer of information from one person or place to another. Take for example, a speaker talking to an audience in an auditorium. If he and the audience speak the same language, then some sort of transfer of information will take place.

The mechanisms for transferring thoughts into words, vocalizing the words into acoustic sounds, hearing these sounds, interpreting them as words, and converting the words back into thoughts, can be considered to be a communication system. This system is perhaps the most familiar of all such systems, however, it also contains the basic elements of a general digital communication system.

The speaker can be considered to be the *transmitter* of information. His thoughts are the *information source*; he will try to communicate specific thoughts to the audience. Each member of the audience is a *receiver* of information. If each member of the audience retains the received information rather than relaying it on to another, then each member is also an *information user*. The air through which the acoustic speech wave

propagates is called the *communication channel.*

The primary function of the transmitter is to prepare any message from the information source for transmission over the communication channel. The primary function of the receiver is to convert the channel output to a form that is acceptable to the user.

It is possible that an ambient sound, such as an air conditioner, is present in the auditorium. In addition, a door may be occasionally opened which allows the outside clamor of traffic to be audible. Both of these noises, present in the communication channel, interfere with the audience's listening ability. The first noise is always present, while the second is a pulse of short duration. If the duration is short enough, it is sometimes classified as impulsive noise. An example might be the blast of a horn.

A general communication system, like the above example, is also subject to what is called *channel noise.* Two types of noise will be considered: additive *random noise*, and additive *impulsive noise.* In the example, the noise of the air conditioner might be modelled as random noise, while the traffic sounds would be impulsive noise.

Strictly speaking, it is common for the receiver and transmitter mechanisms to contribute both random and impulsive noise. However, it is usually easier to view this noise as contributed by the communication channel. An example of this will be the random quantization noise introduced in the transmitter by a communication system that sends real numbers to a user.

For this discussion, only *point-to-point* communication systems will be considered. A point-to-point system in the speaker example would imply that there is one speaker and only one member in the audience.

In addition, only *digital communication systems* will be considered. Loosely, any system which at some point in the communication path uses a finite alphabet at discrete time instances to represent the information source will be considered to be a digital communication system.

The above system is digital in the sense that the speaker's thoughts must be converted into words. Since the number of letters in each word and the number of letters are both finite, the number of words must be finite, and the communication is digital.

So far, the main elements of a digital communication system are the transmitter, the receiver, and the communication channel complete with channel noise. The information source is usually separated from the actual transmitting mechanisms, as the user is separated from the receiver mechanisms. These mechanisms are also categorized into different elements, which are discussed in the next section.

Basic Transmitter and Receiver Elements

The basic elements of the transmitter include a *source coder*, a *channel coder*, and a *modulator*. The receiver attempts to "undo" the coding of the transmitter, and so it consists of a *demodulator*, a *channel decoder*, and a *source decoder*. Figure II.1 depicts these elements in their usual order in relation to the information source, the user, and the channel.

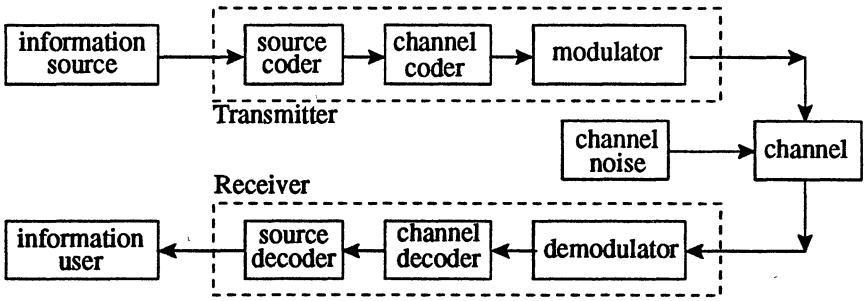


Figure II.1: Basic Digital Communication System Diagram

Consider, again, the speaker example. Figure II.2 illustrates this example for a specific message that needs to be communicated to the listener. Using this example, the primary functions of the transmitter and receiver can be explained.

The rudimentary function of the source coder is to reduce the messages from the information source. In the example, the speaker has a "marvelous, grandiose apperception", and he wants to inform the listener. Since "marvelous, grandiose apperception" is long and he has but a brief instant to talk, he chooses to tell the listener that he has a "good idea". The speaker has just performed the function of source coding. The original thought has been reduced to only two short words.

Of course, the original message has been altered. "Good idea" does not convey the same exact meaning as "marvelous, grandiose apperception". However, since there was just a short time to talk, "good idea" conveys the basic content of the original message and is an acceptable compromise. It is often the case that such compromises must be made by the source coder.

The primary purpose of the channel coder is to combat the channel noise and to compensate for any constraints on the channel. In this example, the speaker does not channel code the message, "good idea", to a large extent. Suppose, he merely emphasizes and extends the vowel sound in the word "good".

The modulator has the task of converting the channel coder output to a form that is suitable for transmission over the communication channel. Since the channel is an open acoustic channel, the words are spoken. (A modulator for a visual channel might use sign language.)

Now suppose that at the instant that the word "good" is spoken, the auditorium door is opened and a the sound of truck horn disrupts the communication channel. The speaker continues with "idea" and the listener is faced with the task of determining what message the speaker is trying to communicate.

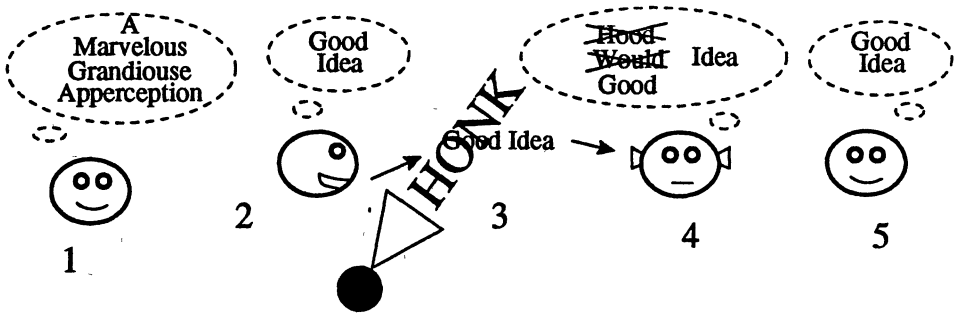


Figure II.2: Speaker-Listener Example

The listener hears the garbled message from the speaker. Converting these sounds back into words or approximations of words is the demodulator's job. Suppose the listener hears the word "idea" clearly, and he also makes out the middle and last parts of the word "good". The function of the channel decoder is performed when the listener thinks about what he heard and decides that the first word spoken was probably "good". Despite the channel interruption, the correct message is interpreted.

Finally, the listener source-decodes this message by putting the two words together and arriving at a mental picture of what they mean. Spoken communication is actually very complicated. This example has been simplified significantly, since different words mean different things to different people. Also, different voice inflections can change the meaning of a word entirely. However, even though this example has been simplified, all the basic elements of a digital communication system are present.

The remainder of this section looks at each element in a bit more detail by introducing examples and terms which are relevant to later chapters of this discussion.

Information Sources

In general, an information source will be considered to be anything that outputs a stream of symbols over time. Both continuous and discrete streams will be considered; a discrete stream being a sequence of symbols that occur at distinct time instances.

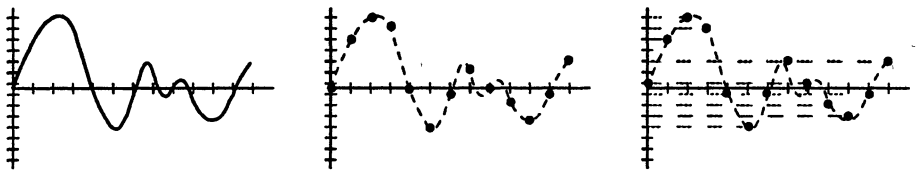
The symbol alphabet can be either infinite or finite. When considering an infinite symbol alphabet, it will be assumed that the symbols are taken from a set like the field of real numbers. Other infinite source alphabets exist, but this discussion does not consider them. Finite source alphabets are easier to handle, and no restrictions will be put on them. In light of this, there are three types of information sources: *waveform* sources, *continuous* sources, and *discrete* sources.

A waveform source is continuous in amplitude and time; meaning, it outputs a continuous stream of symbols drawn from an infinite alphabet. A waveform source can be thought of as an analog signal. Such a source can be modulated directly. For example, voice or music modulated by common AM and FM radio stations. However, this discussion is concerned with digital communication systems, so a waveform source must be processed for use in a digital system.

A continuous source is assumed to be continuous in amplitude, but discrete in time. For example, a sampled analog signal is a common continuous source. Any sequence of real numbers is a continuous source. For a digital system, a continuous source must also be processed. (Modulators that accept continuous amplitude input data will not be considered.)

Finally, a discrete source outputs a sequence of symbols which are taken from a finite alphabet. A sampled analog waveform whose amplitude has been digitized (quantized) is a common discrete source. Another example could be a stream of letters. Discrete sources will also sometimes be called *digital* sources.

Figure II.3 illustrates a waveform source, the sampled continuous version, and a quantized, discrete source. The processes of sampling and quantization convert waveform sources to continuous sources and continuous sources to discrete sources, respectively. These processes fall under the category of source coding and will be discussed in greater detail later in this chapter.



(a) Waveform Source (b) Continuous Source (c) Discrete Source

Figure II.3: Examples of Information Sources
(a) Waveform, (b) Continuous, (c) Discrete.

The process of communication is fundamentally random. That is, if the user already knows the next output of the information source, then no communication will take place. Because of this, it is common to model an information source as a random process.

For example, associated with a discrete source consisting of the symbols $\{a_0, a_1, \ldots, a_{J-1}\}$, is a probability distribution **p**. If this source is a discrete *memoryless* source, then the probability that a source output is a particular symbol at some given time

is independent of any previous symbol outputs. In this case, the probability distribution for the source can be given by, $\mathbf{p} = \{p(a_0), p(a_1), \ldots, p(a_{J-1})\}$, where $p(a_i)$ is the probability of source output symbol a_i.

Similarly, a continuous source can be modelled with a continuous probability distribution, with output symbols occurring at discrete time instances. A waveform source can be viewed as a continuous random process.

Communication Channels

The communication channel provides the link between the information source (transmitter) and the information user (receiver). It might be an acoustic free-space channel as in the speaker example, or it could be a twisted pair of wires that carry an electrical signal. Other examples include guided acoustic, electromagnetic, and optical waveguides. Free space radio and underwater channels are also common. The transfer of computer data to a storage device is a fast growing form of communication. The channel in this case consists of some sort of magnetic or optical media which holds the digital data.

Communication channels will be referred to in three ways, depending upon what sort of input symbol stream they accept. There are *waveform* channels, *continuous* channels, and *discrete* channels. A possible point of confusion is that a single communication system may contain all three types. This is because it is convenient to include transmitter and receiver elements into what can be called a *combined channel.* Figure II.4 depicts an example that combines the modulator and demodulator with a waveform channel to create a discrete channel.

In this case, the output of the modulator is a waveform. For example, it might be a binary pulse train modulated onto a high frequency carrier. The waveform channel is subjected to noise such as interference from other radio waves. The input to the modu-

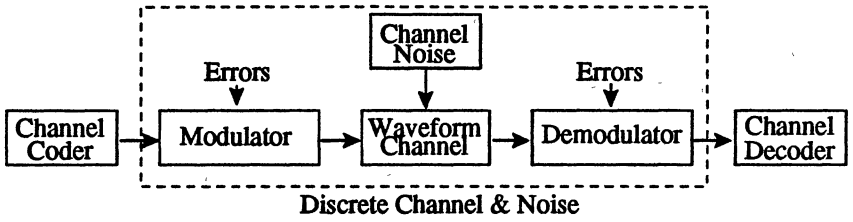


Figure II.4: Combined Discrete-Waveform Channel

lator for this example is a sequence of binary digits (bits). By combining the modulator, the demodulator, and the waveform channel, a single discrete channel can be formed. The modulator and demodulator might be susceptible to equipment errors which will contribute to the combined channel noise of the discrete channel.

Using combined channels is convenient when designing a communication system. It allows the designers to look at the transmitter and receiver elements separately. In this discussion, only discrete channels will be considered. For this reason, modulators and demodulators are not discussed.

A communication channel is further categorized by the types of noise that are present. A *noiseless* channel is not subjected to any disturbances, and thus it makes no errors. A *memoryless* channel implies that the transmission of a given symbol is independent of the transmission of any previous symbols. Thus, the probability of correctly receiving a given transmitted symbol at a given time is independent of any previous transmissions. An *unconstrained* channel will allow input symbols to be transmitted in any arbitrary order.

Similar to information sources, communication channels can be represented with a probabilistic model. One of the most common models for a binary, discrete memoryless channel is the *Binary Symmetric Channel*, (BSC), shown in Figure II.5.

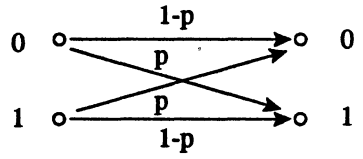


Figure II.5: The Binary Symmetric Channel

The possible input and output symbols for the BSC are the same: binary digits. From the figure, p *is* the *transition probability* of receiving a one given that a zero was transmitted. Since the channel is symmetric, this is equal to the probability of receiving a zero given that a one was transmitted. Thus, p is the probability of a channel error; and $1-p$ is the probability of no channel error. A general discrete memoryless channel with J possible input symbols and K possible output symbols can be represented by a $J \times K$ *transition probability matrix*, $Q = [q_{jk}]$, with q_{jk} being the probability of receiving symbol k given that source symbol j was transmitted. For the BSC, this matrix is given by

$$Q_{BSC} = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}.$$

Another type of channel is called a *burst* channel. This type of channel is subjected to impulsive noise which causes "bursts" of errors. For example, a compact disc can be considered as a communication channel. Any scratch on the disc will cause a burst of errors. Note that a burst channel is not memoryless.

Source Coder

A code for an information source can be considered to be a representation of that source. This representation could be a waveform, a sequence of real numbers, or some discrete representation. For digital communication systems, a discrete representation is required. Since binary representations are the most convenient and common discrete representation, this discussion will concentrate on symbols represented by bits.

As mentioned earlier, the primary purpose of a source coder is that of data reduction. A given source generates symbols at a particular rate. If the source is discrete with symbols $\{a_0, a_1, \ldots, a_{J-1}\}$ and probability distribution $\boldsymbol{p}$, and a particular code uses l_i bits to represent the i^{th} symbol; then the average length of a given source symbol is given by, [Bla87],

$$\bar{l} = \sum_{i=0}^{J-1} p_i l_i. \qquad (II.1)$$

Assuming that the source generates these symbols at a certain rate, R_{sym}, then the *data rate*, $R_{sym}\bar{l}$, is the average number of bits per second required to represent the source by the given code. The source coder attempts to minimize the data rate, while still preserving the original information content of the source.

What is needed is a function that measures the information content of a source. This function is the *entropy function*. The entropy for a discrete memoryless source can be defined as follows:

DEFINITION II.1: Given a discrete memoryless source with probability distribution **p**, then the *entropy* of the source is given by

$$H(\mathbf{p}) = \sum_{i=0}^{J-1} p_i \log_2\left(\frac{1}{p_i}\right), \qquad (II.2)$$

where p_i is the probability of the i^{th} symbol. (Since a base 2 logarithm is used, the entropy is measured in bits.)

The entropy can be thought of as the lower limit on the average number of bits required to represent the information of a given source. A goal of the source coder is to encode the information source so that the average number of bits used to represent the source is equal to the entropy. Thus, the data rate equals the *information rate.*

Following the terminology found in Blahut, [Bla87] & [Bla90], source codes can be divided into two categories: *data compaction codes* and *data compression codes.* Both types of codes will be discussed, however, data compression codes play a much more important role in this document.

Data Compaction Codes

A data compaction code reduces the data rate of a given representation without reducing the information content of the source; i.e. the entropy of the output of a data compaction codes equals the entropy of the input.

An example of a data compaction code for a discrete memoryless source is shown in Figure II.6. The source is a child that is playing a musical instrument that can produce 7 notes: A, B, C, D, E, F, G. The child plays 2 notes per second.

Two representations of the source will be considered. The first code represents the notes with binary 3-tuples while the second code uses binary tuples of variable length. The entropy stays the same for both codes, however, the data rate of the second code is lower than the data rate of the first. The process of converting the first representation into the second is a source coding operation, and the second representation is a data compaction code.

	Note		C_1	C_2
a_0	A	$p_0 = 2/7$	001	00
a_1	B	$p_1 = 1/14$	010	1100
a_2	C	$p_2 = 1/14$	011	1101
a_3	D	$p_3 = 1/7$	100	01
a_4	E	$p_4 = 2/7$	101	10
a_5	F	$p_5 = 1/14$	110	1110
a_6	G	$p_6 = 1/14$	111	1111

$R_{sym} = 2$ symbols/sec.

$\overline{L}_{C1} = 3.0$ bits $\qquad$ $\overline{L}_{C2} = 2.57$ bits

$H(p) = 2.52$ bits

Figure II.6: Example of a Data Compaction Code

A second example of a data compaction code is the proper sampling of a bandwidth limited signal. Suppose that *B* is the highest frequency present in a baseband waveform source. By the Nyquist Sampling Theorem, if the source is sampled at a rate greater than or equal to *2B* (usually greater than *2B*), then the original waveform can be exactly recovered from its samples.

The original representation required an uncountably infinite number of real numbers, while the second merely requires a countable number of real numbers. However, both representations still require an infinite data rate, since they have infinite entropy. It requires an infinite number of bits to represent any real number precisely. Lowering the entropy by throwing out information is the subject of data compression codes.

Data Compression Codes

A data compression code reduces the data rate of a source by reducing the information rate. For the case where an analog waveform is sampled, the data rate can be made finite by rounding or quantizing the samples to a finite set of values. This set of finite values can be considered to be a discrete source, and it has a finite entropy.

Inherent in the process of throwing out information is the addition of *distortion.* The quantized samples do not exactly equal the original samples. Once quantized, the information that was lost can never be recovered.

A key goal in data compression coding is to reduce the information rate to an acceptable level while minimizing the distortion. Of course, there is a trade-off. Higher rates allow for lower levels of distortion and lower rates require a higher level of distortion. The study of this trade-off is the branch of information theory called *rate-distortion theory.*

There are many methods for data compression; among the most popular for video and voice communication systems are the transform and predictive coding techniques. However, a common class of compression methods are of utmost importance for this discussion: uniform and optimal mean square scalar quantization.

A *scalar quantizer* maps a continuous variable into a discrete variable. Let x be a real valued variable with probability density function, $p_x(\lambda)$, and define $\{t_1, t_2, \ldots, t_{L+1}\}$ to be a set of increasing real numbers (called *transition levels*) with t_1 and t_{L+1} equal to the minimum and maximum values of x, respectively. Let x_q denote the quantized value. Denote the set of all possible quantized values (called the *reconstruction levels*) by $\{r_1, r_2, \ldots, r_L\}$.

Then a scalar quantizer is defined by a map, $\phi:\mathbf{R} \rightarrow \{r_1, \ldots, r_L\}$, such that if $x \in [t_k, t_{k+1})$, then $x_q = \phi(x) = r_k$. Figure II.7 depicts the input-output relationship for a quantizer with eight reconstruction levels. The quantization error, $e = x - x_q$, is also shown as a function of x.

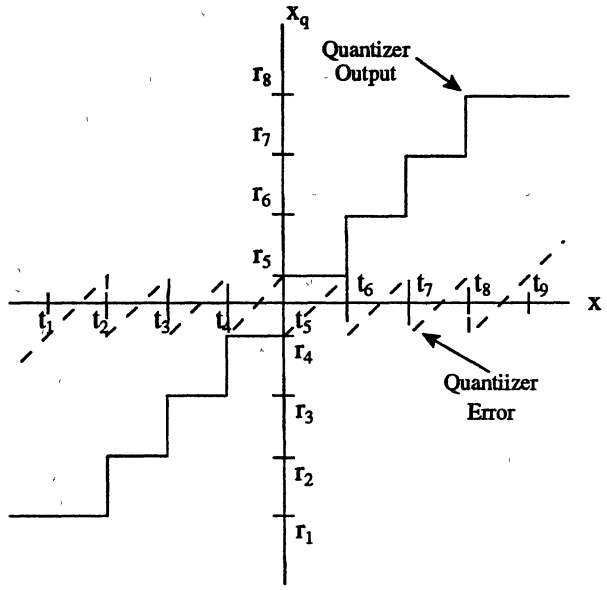


Figure II.7: An Eight Level Scalar Quantizer

An optimum mean square quantizer or *Lloyd-Max (LM) quantizer*, [Llo82], [Max60], minimizes the mean square error (MSE) for a given number of quantization levels. Thus, the LM quantizer finds the sets $\{r_i\}$ and $\{t_i\}$ such that

$$MSE = \sum_{i=1}^{L} \int_{t_i}^{t_{i+1}} (\lambda - r_i)^2 p_x(\lambda) d\lambda, \qquad (III.1)$$

is minimum. This minimization is usually found by using an iterative method, and the results are widely tabulated for different density functions, [Jai89], [Jay84].

When the probability density function of x is uniform, then the LM quantizer becomes a *uniform quantizer*. A uniform quantizer has equal intervals between the transition levels. Uniform quantizers have also been designed for non-uniform densities. These quantizers have a larger MSE than a LM quantizer, however they are easier to implement.

The *signal-to-noise ratio* at the output a quantizer can be defined as

$$SNR = \frac{\sigma^2}{MSE},$$

where σ^2 is the variance of the input signal. Clearly, more reconstruction levels yield a higher SNR.

Channel Coder

The primary purpose of the channel coder is to combat channel noise and constraints. This is accomplished by designing *data translation* and *data transmission* codes, respectively.

Many times a communication channel has constraints. For example, a binary channel may not allow runs of zeros and/or ones greater than some integer. Such a channel is called a *run-length limited* channel. A data translation code for such a channel would convert an unconstrained binary input stream into a stream that is run-limited to match the channel.

Thus, a data translation code can be combined with a constrained discrete channel to form a combined unconstrained discrete channel, schematically shown in Figure II.8.

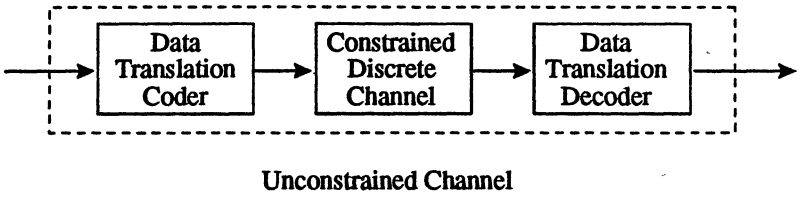


Figure II.8: An Unconstrained Channel Using a Data Translation Code

Channel constraints and data translation codes are not of great importance to this document; and for this reason, they will not be discussed any further. An unconstrained channel will always be assumed.

Of central importance to this document are data transmission codes or *error correction codes*. These codes fall into two categories: *block codes* and *tree codes*. Tree codes are not discussed here; however, block codes are discussed in length. In fact all of Chapter III is devoted to the study of linear block codes.

CHAPTER III

ERROR CORRECTION CODES

The study of error correction codes is a vast and complex topic. It usually requires a strong background in abstract algebra. However, the codes investigated in this document are based upon the real and complex fields, so little algebra, other than the matrix algebra review in Appendix A, is needed. The main purpose for this section is to present some of the basic concepts of a class of error correcting codes, called linear block codes, which are pertinent to the real number (RN) codes studied in this document. Detailed treatment of traditional error correcting codes can be found in [Bla83], [Ber68], [Mac77], and [Pet72]. [Lid84] is also a good reference which views error correction codes as an application of abstract algebra.

Traditional block codes are based upon fields which have a finite number of elements, called *finite fields*. RN codes are based upon the familiar real and complex fields, which have an uncountably infinite number of elements. Any field with an infinite number of elements is called an *infinite field.*

A theoretical presentation of fields is well beyond the scope of this discussion, so instead, several examples and results of finite and infinite fields are presented. Finite fields are introduced so that traditional linear block codes will make more sense to the reader unfamiliar with this topic. Several comparisons can be made between finite field (FF) codes and RN codes. By being familiar with the traditional codes, the reader can appreciate the similarities and differences between FF and RN codes.

Fields

Loosely speaking, a field is a set of elements which can be added, subtracted, multiplied, and divided. A more formal definition requires a few preliminary definitions.

DEFINITION III.1: If S is a set, then a *binary operation* * on S is a function that assigns to each ordered pair (s_1, s_2) of elements in S, another element, $s_3 \in S$. This mapping is denoted by $s_3 = s_1 * s_2$. (Note: + and • are common symbols for binary operations.)

A binary operation is *commutative* when $s_1 * s_2 = s_2 * s_1, \quad \forall s_1, s_2 \in S$. A binary operation is *associative* if $(s_1 * s_2) * s_3 = s_1 * (s_2 * s_3) \quad \forall s_1, s_2, s_3 \in S$. An *identity* with respect to * is an element $e \in S$, such that $s * e = e * s = s, \quad \forall s \in S$. An *inverse* for an element s is an element $s^{-1} \in S$ such that $s * s^{-1} = s^{-1} * s = e$.

Combining these concepts lead to the definition of a group.

DEFINITION III.2: A set G together with an associative binary operation is called a *group*, denoted $(G, *)$, if it has the following properties: G has an identity, and every element of G has an inverse.

It can be shown that inverses and the identity for a group are unique. A group that has a commutative binary operation is called an *abelian group*.

It is customary, that if a group's binary operation is addition (denoted by +), then the identity is the zero element, 0, and the inverse of an element $g \in G$ is denoted by $-g$.

Similarly, the identity for a multiplicative group (multiplication is denoted by • or by juxtaposition) is denoted by 1. The multiplicative inverse is given as originally defined, i.e. g^{-1}.

A field can now be defined as follows:

DEFINITION III.3: A *field* is a set F together with two binary operations, addition and multiplication, such that the following hold:

1) $(F,+)$ is an abelian group.
2) The set of nonzero elements of F is an abelian group under multiplication.
3) (Distributive Law) $(f_1+f_2)f_3 = f_1f_3+f_2f_3$ holds $\forall f_1,f_2,f_3 \in F$.

Some examples and results of infinite and finite fields are now presented.

Infinite Fields

Possibly the two most common infinite fields are the fields of real and complex numbers, denoted by **R** and **C**, respectively. Another common example is the field of *rational numbers*, **Q**, defined by

$$\mathbf{Q} = \left\{ \frac{m}{n} \mid m,n \in \mathbf{Z} \right\},$$

where **Z** is the set of integers. **Z** itself is not a field since there are no multiplicative inverses. For example, there is no $z \in \mathbf{Z}$ such that $4z = 1$.

It is assumed that the reader is familiar with the real and complex fields, along with vector spaces over these fields. If not, Appendix A contains a brief review of vector spaces over the real field. The extension of the results in Appendix A to the complex field is not difficult.

Since the real and complex fields are the only infinite fields that are of interest in this discussion, no other infinite fields will be discussed. Instead, some examples of finite fields will be presented.

Finite Fields

A field **F** is finite if the number of elements in **F**, denoted $|\mathbf{F}|$, is finite. $|\mathbf{F}|$ is called the *order of* **F**. Probably the most common finite field is the binary field, denoted by $\mathbf{F}_2 = \{0, 1\}$. Addition and multiplication in the binary field is performed modulo 2. The standard addition and multiplication tables for $\mathbf{F}_2$ are given in Figure III.1.

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

Figure III.1: Addition and Multiplication Tables For the Binary Field

Other finite fields can be defined in a similar fashion. Let

$$\mathbf{Z}_n = \{0, 1, \ldots, n-1\},$$

then if $n = p$, where p denotes a prime number, $\mathbf{Z}_p$ with *mod p* arithmetic is a field. As an example, consider $p = 7$. Figure III.2 depicts the addition and multiplication tables for $\mathbf{F}_7$.

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

·	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

Figure III.2: Addition and Multiplication Tables For $\mathbf{F}_7$.

Many fields of interest do not have a prime number of elements, and so their construction is not as simple as that of $\mathbf{F}_p$. Of special interest are those fields whose order is a power of 2. For example, a field with 2^n elements can be conveniently represented with binary n-tuples. Alternately, binary n-tuples can be given the structure of a field. For this reason, traditional error correction codes use fields whose order is a power of two.

A general result concerning finite fields is that for any finite field, the number of elements is a power of a prime. In addition, there exists a field of order p^n, with p a prime, [Lid84]. The prime, p, is said to be the *characteristic* of the field.

In addition, up to isomorphism, this field is unique. A *field isomorphism* is a one-to-one and onto mapping that preserves the additive and multiplicative structure of the field. Loosely speaking, two fields $\mathbf{F}_1$ and $\mathbf{F}_2$ are isomorphic if $\mathbf{F}_1$ can be transformed into $\mathbf{F}_2$ merely by renaming its elements with elements from $\mathbf{F}_2$.

Constructing fields of order p^n requires that n-tuples be represented by degree *n-1* polynomials with coefficients from $\mathbf{F}_p$. Addition and multiplication of these polynomials is then taken modulo some prime polynomial of degree *n*. This construction is directly analogous to the construction of $\mathbf{F}_p$ from $\mathbf{Z}_p$, only polynomials are the field elements, instead of integers.

A detailed discussion of this construction is beyond the scope of this section, so an example is presented as a substitute. This example constructs two "versions" of the field $\mathbf{F}_8$, using two degree 3 binary polynomials:

$$f_1(x) = x^3 + x^2 + 1.$$

$$f_2(x) = x^3 + x + 1.$$

Another result of finite field theory is that the non-zero elements of a finite field can be represented as powers of a single generating element. (This element is called a *primitive element.*) Multiplication in the finite field is simplified by using powers of a primitive element to represent the field elements.

In the example, both $f_1(x)$ and $f_2(x)$ are special prime polynomials called *primitive* polynomials. Since $f_1(x)$ and $f_2(x)$ are primitive polynomials, the field element represented by x is a primitive element. (See Figure III.3) By selecting a primitive polynomial, the construction of the field is less complicated. Also, since primitive polynomials exist for every finite field, limiting oneself to such polynomials does not introduce too many restrictions.

Figure III.3 depicts four different representations for each of the two fields $(\mathbf{F}_8)_1$ and $(\mathbf{F}_8)_2$. The first representation is as integers, the second as binary 3-tuples, the third as polynomials, and the fourth as powers of a primitive element. Figures III.4 and III.5 show the addition and multiplication tables for these two representations of $\mathbf{F}_8$.

Z_8	$(Z_2)^3$	$Z_2[x]/f_1(x)$	α^n
0	000	0	0
1	001	1	α^0
2	010	x	α^1
3	100	x^2	α^2
4	101	x^2+1	α^3
5	111	x^2+x+1	α^4
6	011	$x+1$	α^5
7	110	x^2+x	α^6

(a) $(F_8)_1$

Z_8	$(Z_2)^3$	$Z_2[x]/f_2(x)$	γ^n
0	000	0	0
1	001	1	γ^0
2	010	x	γ^1
3	100	x^2	γ^2
4	011	$x+1$	γ^3
5	110	x^2+x	γ^4
6	111	x^2+x+1	γ^5
7	101	x^2+1	γ^6

(b) $(F_8)_2$

Figure III.3: Two Field Isomorphisms of $\mathbf{F}_8$

An isomorphism between $(\mathbf{F}_8)_1$ and $(\mathbf{F}_8)_2$ is given by:

$$(\mathbf{F}_8)_1 \leftrightarrow (\mathbf{F}_8)_2$$

$$0 \leftrightarrow 0$$

$$1 \leftrightarrow 1$$

$$2 \leftrightarrow 6$$

$$3 \leftrightarrow 4$$

$$4 \leftrightarrow 2$$

$$5 \leftrightarrow 7$$

$$6 \leftrightarrow 5$$

$$7 \leftrightarrow 3$$

Alternately, since the non-zero elements of both fields are described as powers of the primitive elements α and γ, the isomorphism $\phi:(\mathbf{F}_8)_1 \rightarrow (\mathbf{F}_8)_2$ can be specified by

$$\phi(\alpha) = \gamma^5.$$

Powers of α and γ are taken mod 7. The construction of larger fields of characteristic 2, can be performed in a similar fashion.

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	0	6	4	3	7	2	5
2	2	6	0	7	5	4	1	3
3	3	4	7	0	1	6	5	2
4	4	3	5	1	0	2	7	6
5	5	7	4	6	2	0	3	1
6	6	2	1	5	7	3	0	4
7	7	5	3	2	6	1	4	0

·	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	3	4	5	6	7	1
3	0	3	4	5	6	7	1	2
4	0	4	5	6	7	1	2	3
5	0	5	6	7	1	2	3	4
6	0	6	7	1	2	3	4	5
7	0	7	1	2	3	4	5	6

Figure III.4: Addition and Multiplication Tables For $(F_8)_1$

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	0	4	7	2	6	5	3
2	2	4	0	5	1	3	7	6
3	3	7	5	0	6	2	4	1
4	4	2	1	6	0	7	3	5
5	5	6	3	2	7	0	1	4
6	6	5	7	4	3	1	0	2
7	7	3	6	1	5	4	2	0

·	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	3	4	5	6	7	1
3	0	3	4	5	6	7	1	2
4	0	4	5	6	7	1	2	3
5	0	5	6	7	1	2	3	4
6	0	6	7	1	2	3	4	5
7	0	7	1	2	3	4	5	6

Figure III.5: Addition and Multiplication Tables For $(F_8)_2$

Linear Block Codes

Linear block codes are a special type of data transmission codes. They are perhaps the easiest and most widely used type of data transmission codes.

As stated in Chapter II, a code is merely a representation of a data symbol. The source coder attempts to reduce the data rate of given stream of symbols by removing the redundancy in the input stream. On the other hand, the channel coder protects against channel noise by adding redundancy, thereby increasing the data rate.

For example, an *(N,K) block code* maps a block of K symbols into a block of N symbols, where $N > K$. A linear block code implies that the map is linear, however, not any linear mapping will do. In the next section, some of the general concepts of linear block codes are presented; this will help distinguish good mappings from poor mappings.

General Concepts

Before defining linear block codes, consider the simplest binary error correcting code. This code is a (3,1) repeat code. That is, each information symbol (information symbols will be considered to be those symbols at the input to the channel coder) is repeated three times. Thus,

$$0 \rightarrow 000,$$

$$1 \rightarrow 111.$$

Formally, this mapping is a one-to-one mapping from $\mathbf{F}_2$ into $(\mathbf{F}_2)^3$. The two code-words are both three dimensional and are depicted in Figure III.6.

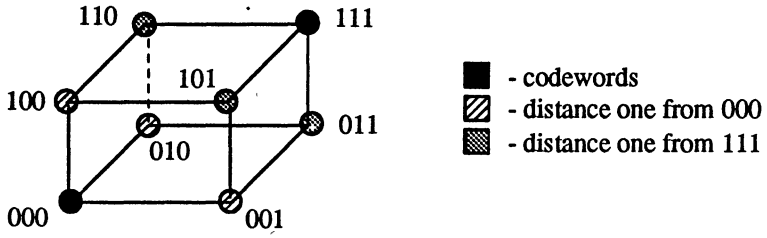


Figure III.6: Codeword Space of (3,1) Repeat Code.

Since the transmission channel is noisy, any of the vertices of the cube is a possible received vector. Clearly, this code can correct a single error. The decoder merely chooses the codeword which is "closest" to the received vector. If two bit errors were present, then a decoding error will be made.

For a memoryless channel like the BSC, choosing the "closest" codeword corresponds to choosing the most likely codeword, provided that the probability of an error is less than one half and each codeword is equally likely. In general, this type of decoding will be called *maximum likelihood decoding*. It is the method of choice for minimizing the probability of a decoding error.

The number of errors that a code can correct is determined by the minimum distance of the code. The distance measure is the standard *Hamming distance*, denoted by $d_{Ham}(c_1, c_2)$, where c_1 and c_2 are both codewords. It is defined as the number of elements in which the two codeword vectors are different. The *minimum distance*, $d_{\min}$, is the minimum distance between any two codewords.

For the simple (3,1) repeat code, $d_{\min} = 3$. Because *000* and *111* differ in 3 places, the Hamming distance is 3. This is the minimum distance for the code, since there are only two codewords.

The Hamming distance is traditional for vector spaces over finite fields. However, for real number vector spaces, the standard Euclidean distance is more natural. It will be seen that a linear block code forms a subspace of $\mathbf{F}^N$. Thus for an RN code, the Euclidean distance between two codewords can be made arbitrarily close to zero. Therefore, no natural minimum distance measure using the Euclidean distance seems applicable.

One can still define the minimum distance for an RN code using the Hamming distance, as is traditional for FF codes; however, for this discussion, an alternate definition will be given. First, a formal definition of linear block codes is needed.

An (N,K) linear block code can be defined by a mapping from $\mathbf{F}^K$ into $\mathbf{F}^N$, where $\mathbf{F}$ is a specified field. Let this mapping be denoted by $G:\mathbf{F}^K \rightarrow \mathbf{F}^N$. A vector $c \in \mathbf{F}^N$ will be called a *codeword*, if there exists a $d \in \mathbf{F}^K$ such that $G(d) = c$. The vector d will be called the *information word* or *data word*. This mapping must be one-to-one so that associated with every length N codeword there is a unique information word.

Since G is a linear map, it can be represented by a matrix; thus let $G \in \mathbf{F}^{N \times K}$. G is called the *generator matrix*. The encoding equation is then given by[1]

$$c = Gd. \qquad (III.1)$$

The code itself is given by $C = Im(G)$ and is a subspace of $\mathbf{F}^N$. (See Definition A.6 and Theorem A.7.) For this reason, it will commonly be called the *codespace*. It consists of all $c \in \mathbf{F}^N$ such that $c = Gd$ for some $d \in \mathbf{F}^K$. Since G is one-to-one, this implies that $rank(G) = K$ and $dim(C) = K$.

The ratio K/N is called the *rate* of the code. A lower code rate reflects that more redundancy is added by the channel coder.

An alternate definition for a linear block code uses a full rank $N \times (N-K)$ matrix H, called the *parity check matrix*.

DEFINITION III.4: An (N,K) linear block code is the set of vectors $c \in C \subset \mathbf{F}^N$, such that $H^T c = 0$. (In the case where $\mathbf{F}$ is the complex field, H^T should denote the conjugate transpose.)

---

1 The author differs from the usual error correction code notation which uses row vectors. Since real and complex codes are the main focus of this document, the traditional notation for matrix computations has been adopted

For this definition, $C = Ker(H^T)$. (See Definition A.8 and Theorem A.9.) In general, both a parity check matrix and a generator matrix will be specified for a given code. Combining the two definitions leads to the following orthogonality condition:

$$H^T G = 0. \qquad (III.2)$$

Equation (III.2) implies that the columns of H are orthogonal to the columns of G. Since G is full rank, the columns of G form a basis for C. Consequently, the columns of H form a basis for $C^{\perp}$. (See Definition A.11 and footnote.)

Note that for a given G, the parity check matrix, H, is not unique. If $V \in \mathbf{F}^{N-K \times N-K}$ is of full rank, then $H' = HV$ is still a parity check matrix for G. The code defined by H' is said to be *equivalent* to the code defined by H. Similarly, for a given H, G is not unique.

Suppose that $G^T = [I_K \mid P^T]$. Then the parity check matrix can be specified by $H^T = [-P \mid I_K]$, since

$$\begin{aligned} H^T G &= [-P \mid I_K] \begin{bmatrix} I_K \\ P \end{bmatrix} \\ &= -P + P \\ &= 0. \end{aligned}$$

A code of this form is called *systematic*. A systematic code allows for easy decoding since the information word appears directly in the codeword.

During transmission, the codeword might be corrupted by channel errors. Let $r \in \mathbf{F}^N$ denote the received word, and let

$$r = c + e, \qquad (III.3)$$

where $e \in \mathbf{F}^N$ is an error vector. The number of nonzero elements of e, defined here as the *weight* of e, is equal to the number of symbol errors. (Recall that each symbol is a field element.)

The *syndrome vector*, $s \in \mathbf{F}^{N-K}$, is given by

$$\begin{aligned} s &= H^T r \\ &= H^T(c+e) \\ &= H^T(Gd+e) \\ &= H^T e. \end{aligned} \qquad (III.4)$$

The collection of all syndromes forms an $N-K$ dimensional subspace called the *syndrome space*. Note that the syndrome vector is dependent only upon the error vector. Its computation is the first step in the error correction process.

Usually, a code is described by both the number of errors that it can correct and the number of errors it can detect. These parameters depend upon the minimum distance. In turn, the minimum distance can be found from the parity check matrix.

Denote the columns of H^T by

$$H^T = [h_0, h_1, \ldots, h_{N-1}].$$

Let $L = \{l_1, l_2, \ldots, l_\mu\}$ be a set of distinct indices such that $L \subset \{0, \ldots, N-1\}$ and $l_1 < l_2 < \cdots < l_\mu$. L is called an *index set* for the (N,K) code. The number of indices in L is called the *order of* L, and is denoted by $|L|$. Let H_L^T be the matrix formed from the columns of H^T corresponding to the indices of L. (H_L is the matrix formed by retaining the rows corresponding to the indices of L.)

Consider the rank of H_L^T. Since H is an $N \times (N-K)$ matrix, $rank(H_L^T) \leq N-K$ for all L. If $|L| \leq N-K$, then $rank(H_L^T) \leq |L|$.

Let ξ be the maximum number of columns of H^T, such that any set of ξ columns are always independent. Thus,

$$\xi = \max\{\mu \mid rank(H_L^T) = \mu \quad \forall L \text{ with } |L| = \mu\}.$$

Now an alternate definition for the minimum distance of a code C can be given as follows:

DEFINITION III.5: For a given code defined by H, the *minimum distance* is given by,

$$d_{\min} = \xi + 1. \qquad (III.5)$$

It can be shown, that the two definitions of minimum distance are equivalent.

Given the minimum distance, the number of errors a code can correct and detect are given by the following theorem.

THEOREM III.6: If $d_{\min} \geq 2t + \rho + 1$, then the code C can simultaneously correct up to t errors and detect up to ρ additional errors.

PROOF: Show for the worst case where $d_{\min} = 2t + \rho + 1$. In this case

$\xi = d_{\min} - 1 = 2t + \rho$.

Assume t errors at locations $L = \{l_1, \ldots, l_t\}$ with non-zero values $\{e_{l_1}, \ldots, e_{l_t}\}$. The decoder will attempt to find smallest index set and corresponding error values to match the syndrome. Thus the syndrome, s, is uncorrectable if there exists an index set $J = \{j_1, \ldots, j_v\}$ with $L \neq J$ and non-zero error values $\{e'_{j_1}, \ldots, e'_{j_v}\}$ such that

$$s = \sum_{i=1}^{t} e_{l_i} h_{l_i} = \sum_{i=1}^{v} e'_{j_i} h_{j_i}.$$

This would imply that

$$0 = [h_{l_1}, \ldots, h_{l_t}, h_{j_1}, \ldots, h_{j_\nu}] \begin{bmatrix} e_{l_1} \\ \cdot \\ \cdot \\ e_{l_t} \\ -e'_{j_1} \\ \cdot \\ \cdot \\ -e'_{j_\nu} \end{bmatrix}. \qquad (III.6)$$

Since $\xi = 2t + \rho$ with $\rho \geq 0$, then either $\nu > t$ or else all the error values must be zero. Therefore, the code can correct all t-error syndrome patterns.

Now assume a total of $t + \rho$ errors, with $\rho \geq 1$. Since $\xi = 2t + \rho$, then $s = \sum_{i=1}^{t+\rho} e_{l_i} h_{l_i} \neq 0$. Now using the same argument as before, if $\nu > t$ then (III.6) with t replaced with $t + \rho$, does not give a unique solution. However, since the syndrome is non-zero, an error is detected. Thus, this code can correct no more than t errors, while detecting ρ errors.

If more than $t + \rho$ errors are present, then there exists an incorrect t-error pattern with the same syndrome. In this case, the decoder attempts to correct the error pattern and fails. Thus, the code can detect no more than ρ errors.

■

In this document, it will be assumed that $\rho = 0$. Thus,

$$d_{\min} \geq 2t + 1.$$

The maximum number of errors that a code can correct is given by

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor, \qquad (III.7)$$

where $\lfloor \cdot \rfloor$ is an operator that returns the integer part of the operand.

Since $\xi \leq N-K$, $d_{\min} \leq 1+N-K$. This inequality is known as the *Singleton Bound.* A code that satisfies the Singleton bound with equality in known as a *maximum distance code.*

A code can correct a given number of errors whenever the error locations are uniquely determined from the syndrome. This occurs when the number of errors is less than or equal to t. Once these locations are known, the error correction process can be done in two different ways:

1) From the syndrome equations, (III.4), the error values can be determined. Next, using (III.3) the error values can be subtracted from the received word to get the corrected codeword. Finally, the codeword can be used to get the information word.
2) From the encoding equation, (III.1), the rows corresponding to the error locations can be deleted. The resulting system of equations can be used to directly solve for the information word.

The first procedure is well defined since Theorem III.6 guarantees that the error values can be uniquely determined. In addition, since G is of full rank, solving (III.1) with $c \in C$ gives a unique information word.

The second procedure is well defined if the deleted generator matrix is still of full rank. This is guaranteed by the following theorem:

THEOREM III.7: Given a code C with generator matrix G and minimum distance $d_{\min}$, then $rank(G_L) = K \quad \forall$ index sets L, so long as $|L| > N - d_{\min}$.

PROOF: Show for the worst case where $|L| = N - d_{\min} + 1$. Know $G_L \in \mathbf{F}^{|L| \times K}$. By using the Singleton Bound,

$$|L| = N - (d_{\min} - 1)$$
$$\geq N - (N - K)$$
$$= K.$$

Thus, $|L| \geq K$ and the system of equations $c_L = G_L d$ is either square or overdetermined.

$\therefore rank(G_L) \leq K.$

Assume that $rank(G_L) < K$. Then there exists $d_1, d_2 \in \mathbf{F}^K$ such that

$$c_L = G_L d_1 = G_L d_2.$$

$\Rightarrow$ $d_{Ham}(Gd_1, Gd_2) =$ (number of rows deleted) $= N - |L|$. But $d_{Ham}(Gd_1, Gd_2) \geq d_{\min}$, so $N - |L| \geq d_{\min}$ which is a contradiction. Thus, $rank(G_L) = K$.

■

So far, nothing has been said about how the error locations are found. In general, if the error locations are given by $L = \{l_1, \ldots, l_\mu\}$ with $\mu \leq t$, then a decoding algorithm must determine that $s \in span(h_{l_1}, \ldots, h_{l_\mu})$. Desirable codes not only try to maximize the error correction abilities for a given rate, but they also need to have efficient decoding algorithms. The Reed-Solomon and BCH codes have such an algorithm; for this reason, they are very popular.

Single Error Correcting Codes

Consider the parity check matrix for a single error correcting code given by,

$$H^T = [h_0, \ldots, h_{N-1}]. \qquad (III.8)$$

Since $t = 1$, it is known that $d_{\min} - 1 = 2$. Thus, any 2 columns of H^T are independent. This guarantees that no column is a multiple of another.

Decoding such a code is simple. The syndrome is of the form,

$$s = eh_i,$$

where a single error has occurred in the i^{th} position. The span of each column of the H^T forms a one dimensional subspace of the syndrome space. A syndrome corresponding to an error vector of weight one must lie in one of these subspaces. At most, the decoder must try N subspaces.

Of course, for some codes, such a search is not necessary. Consider the parity check matrix for the binary, single error correcting (7,4) Hamming code.

$$H^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

The rows of H have been ordered such that when interpreted as binary numbers, the rows equal the row number. Since this code is binary, all error values are equal to one. Thus, $s = h_i$, and the syndrome gives the error location as a binary number. Note that the addition of any two rows yields another row. Thus, this code can correct one error and detect no other errors.

Now consider a real number code with parity check matrix of the form (III.8). Again the span of each row is a one dimensional subspace of the syndrome space. These subspaces will be called *error subspaces*. There are N of these subspaces which lie in the syndrome space and intersect only at zero.

Since it is possible to have any given number of linear subspaces in $\mathbf{R}^2$, a single error correcting RN code can be of the form (N, N-2). Unlike the (7,4) binary Hamming code, the syndrome space only needs to be two dimensional. Figure III.7 depicts one possible syndrome space configuration for N = 7.

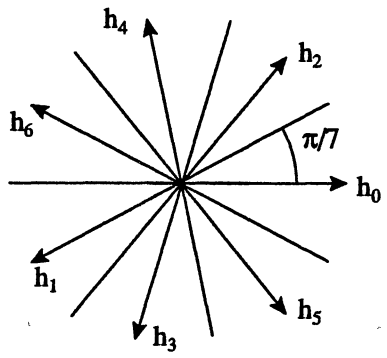


Figure III.7: Syndrome Space for t=1, (7,5) RN Code

In this case, all the error subspaces are equally spaced, with an minimum angle of $\pi/7$. In general, an angle of π/N is possible. If there is no roundoff or quantization noise, then this angle does not matter. However, in the case where such noise is present, the spacing affects the decoding procedure. This topic is discussed in detail in Chapter V.

For this code, an easy decoding method is available since each error subspace has a fixed angle with the x-axis. The inner product of the syndrome with the x-axis will give the desired angle, which in turn gives the error location.

Multiple Error Correcting Codes

The single error correcting Hamming codes were first presented in the late 40's and early 50's, [Sha48], [Ham50]. However, there was a need to correct more than a single error. General codes that would correct multiple errors were not discovered until more than a decade later. These codes were the Bose-Chaudhuri-Hocquenghem (BCH) and

Reed-Solomon (RS) codes, [Bos60], [Hoc59], & [Ree60]. The sharp increase in complexity of the multiple error correcting codes compared to the simple single error correcting codes is responsible for the delay.

Consider the parity check matrix in (III.8). Suppose a t-error correcting code is desired. In the case where $t = 1$, the syndrome space is two or three dimensional. The syndromes are easy to visualize. In addition, only all sets of 2 rows of H need to be independent. As t increases, the code rate decreases and the syndrome space can no longer be visualized. Verifying the minimum distance of such a code by brute force can be computationally infeasible.

Instead of brute force searches, more structured methods are needed. Sound construction rules and decoding algorithms are required. Brute force decoding of a t-error correcting code would require the search over all possible t-dimensional error syndrome subspaces. For large N, this is not feasible.

The RS and BCH codes are a class of highly structured codes with efficient decoding algorithms. The real and complex versions of these codes are discussed in the next chapter.

CHAPTER IV

REAL NUMBER BCH AND RS CODES

Reed-Solomon and BCH codes are popular error correcting codes partly because there is a well defined algorithm that calculates the error locations. This algorithm is a result of the highly structured generator and parity check matrices that exist for these codes.

One way to define RS and BCH codes is through the Discrete Fourier Transform (DFT) matrix. The DFT matrix is very structured. It is symmetric, Vandermonde, and unitary. Since the DFT matrix is very common in many aspects of engineering, using the DFT to define RS and BCH codes is not only illustrative, but also convenient.

The DFT Matrix

The discrete Fourier Transform exists over many fields. For example, DFT matrices exist for the previously mentioned finite fields: $\mathbf{F}_7$ and $\mathbf{F}_8$. In the case where the underlying field is finite, a DFT matrix of size N exists, whenever $N \mid (p^m - 1)$. Thus, for $\mathbf{F}_7$, non-trivial DFT matrices of size 2,3 and 6 exist. For $\mathbf{F}_8$, only a 7×7 DFT matrix exists.

The most common DFT matrix is based upon the complex field. Complex DFT matrices exist for every N. If ω is a primitive N^{th} root of unity, then

$$W_N = \frac{1}{\sqrt{N}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{N-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \omega^0 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}, \qquad (IV.1)$$

$$= [w_0, w_1, \ldots, w_{N-1}],$$

is an $N \times N$ DFT matrix over the complex field, where w_i is the i^{th} column vector. Similar to a primitive element in a finite field, powers of a primitive N^{th} root of unity generate all the N^{th} roots of unity. For example $\omega = e^{-j\frac{2\pi}{N}}$, where $j = \sqrt{-1}$, is a primitive N^{th} root of unity.

A complex DFT matrix is unitary, i.e. $W_N^{-1} = W^H$. The superscript H denotes the complex conjugate transpose of a matrix. In order to avoid any possible confusion with the parity check matrix, henceforth, the notation A^H will be dropped in favor of A^T. If A is complex, it should be understood that A^T is the conjugate transpose. If there is a need, transposition without conjugation will be denoted by $(A^T)^*$. If $v \in \mathbf{C}^N$, then

$$V = W_N v, \qquad (IV.2)$$

is the discrete Fourier Transform of v. Commonly, v is said to be a "time domain" vector, while V is in the "frequency domain". More explicitly, if $V = [V_0, V_1, \ldots, V_{N-1}]^T$ and $v = [v_0, v_1, \ldots, v_{N-1}]^T$, then

$$V_k = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} \omega^{ik} v_i. \qquad (IV.3)$$

V_i is said to be the transform value corresponding to "frequency" ω^i.

Reed-Solomon Codes

Using the complex DFT matrix, a real number (RN) Reed-Solomon (RS) code can be defined as follows:

DEFINITION IV.1: A (N,K) *complex Reed-Solomon code* is the set of all vectors $c \in \mathbf{C}^N$ such that $C_m = 0$ for $m = K, K+1, \ldots, N-1$, where $C = W_N c$.

It is common to refer to $\{\omega^K, \omega^{K+1}, \ldots, \omega^{N-1}\}$ as *parity frequencies* or *zero frequencies*.

Definition IV.1 corresponds to what is found in [Bla90] and [Bla85]; however, any N-K consecutive powers of a primitive element can be used as parity frequencies. For example, $m = 1, \ldots, N-K$ would work just as well. Figures IV.1a and IV.1b depict two possible sets of parity frequencies for a complex Reed-Solomon code, with $\omega = e^{j\frac{2\pi}{N}}$. The frequencies are represented as points on the unit circle in the complex plane.

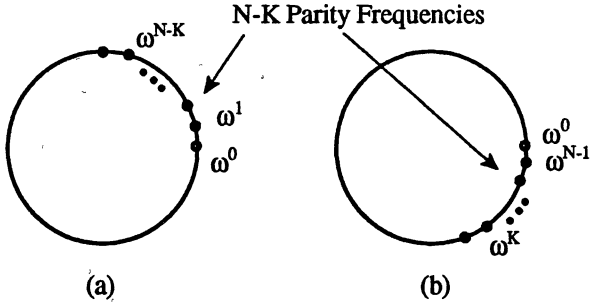


Figure IV.1: Two Possible Parity Frequency Sets

From definitions IV.1 and III.4, it is clear that a parity check matrix can be created from the last N-K rows of W_N. In addition, the first K rows can be used as columns of a generator matrix. Since $W^T W = I_N$, the orthogonality condition of $H^T G = 0$ is satisfied. Using the fact that W_N is symmetric, let

$$W_N = [w_0, \ldots, w_{K-1}, w_K, \ldots, w_{N-1}]$$
$$= [G, H]. \qquad (IV.4)$$

The error correction capabilities for a RS code are specified by the following theorem:

THEOREM IV.2: An (N,K) Reed-Solomon code as given by Definition IV.1, is a maximum distance code, thus $d_{\min} = N - K + 1$.

PROOF: Let the parity check matrix be given by,

$$H^T = \begin{bmatrix} \omega^0 & \omega^K & \ldots & \omega^{K(N-1)} \\ \omega^0 & \omega^{K+1} & \ldots & \omega^{(K+1)(N-1)} \\ . & . & \ldots & . \\ \omega^0 & \omega^{N-1} & \ldots & \omega^{(N-1)(N-1)} \end{bmatrix}.$$

Need to show that any N-K columns are independent. Let

$$L = \{l_1, \ldots l_{N-K}\} \subset [0, 1, \ldots, N-1]$$

be an index set. For $H_L^T \in \mathbf{C}^{N-K \times N-K}$, show H_L^T is of full rank.

Write

$$H_L^T = \begin{bmatrix} (\omega^K)^{l_1} & \cdots & (\omega^K)^{l_{N-K}} \\ . & \cdots & . \\ (\omega^{N-1})^{l_1} & \cdots & (\omega^{N-1})^{l_{N-K}} \end{bmatrix}$$

$$= \begin{bmatrix} \left(\omega^{l_1}\right)^0 & \cdots & \left(\omega^{l_{N-K}}\right)^0 \\ . & \cdots & . \\ \left(\omega^{l_1}\right)^{N-K-1} & \cdots & \left(\omega^{l_{N-K}}\right)^{N-K-1} \end{bmatrix} \begin{bmatrix} \left(\omega^{l_1}\right)^K & \cdots & 0 \\ . & \cdots & . \\ 0 & \cdots & \left(\omega^{l_{N-K}}\right)^K \end{bmatrix}$$

$$= AB.$$

Note that A is a Vandermonde matrix with $\omega^{l_1}, \ldots, \omega^{l_{N-K}}$ distinct and non-zero. Thus, $rank(A) = N-K$. Also, since all diagonal entries of B are non-zero, $rank(B) = N-K$. Thus, $rank(H_L^T) = N-K$.

■

BCH Codes

BCH codes can be defined using the earlier discussion on RS codes. Both the generator and parity check matrices for a Reed-Solomon code are complex. An RS code maps a complex information vector into a complex codeword. By choosing specific parity frequencies, it is possible to design real generator and parity check matrices.

For example, consider a (5,3) RS code. Construct a 5×5 DFT matrix with primitive root $\omega = e^{-j\frac{2\pi}{5}}$. So let

$$W_5 = [w_0, w_1, w_2, w_3, w_4].$$

Then a generator and parity check matrix for the (5,3) RS code can be given by,

$$G = [w_0, w_1, w_4]$$

$$H = [w_2, w_3].$$

Now since $\omega^2 = (\omega^3)^*$ and $\omega^1 = (\omega^4)^*$, then

$$G = [w_0, w_1, w_1^*]$$

$$H = [w_2, w_2^*].$$

Since addition or subtraction of the columns of G does not change $Im(G) = C$, another generator matrix can be specified by

$$G_{\mathbf{R}} = \left[w_0, \frac{w_1 + w_1^*}{\sqrt{2}}, -j\left(\frac{w_1 - w_1^*}{\sqrt{2}} \right) \right]$$

$$= [w_0, \sqrt{2}\Re(w_1), \sqrt{2}\Im(w_1)]$$

where $\Re(w_1)$ and $\Im(w_1)$ denote the real and imaginary parts of w_1, respectively. The factor $\sqrt{2}$, is included for proper normalization, i.e. $G_{\mathbf{R}}^T G_{\mathbf{R}} = I_K$.

Note that the matrix $G_{\mathbf{R}}$ is a real matrix. Similarly, a real parity check matrix is given by

$$H_{\mathbf{R}} = [\sqrt{2}\Re(w_2), \sqrt{2}\Im(w_2)].$$

With these two real matrices, it is now possible to restrict the codespace to be real, merely by restricting the information vector to be real. Thus, if $d \in \mathbf{R}^K$, then $C \subset \mathbf{R}^N$.

Real generator and parity check matrices can be formed only when the parity frequencies are constrained to occur in conjugate pairs. This constraint is called the *conjugacy constraint* and the resulting code is a BCH code. A formal definition can be given as follows:

DEFINITION IV.3: An (N,K) *BCH code* is an (N,K) RS code that obeys the conjugacy constraint: i.e., if ω^m is a parity frequency, then $(\omega^m)^*$ is also a parity frequency.

Since a BCH code is a Reed-Solomon code, the error correction capabilities will be the same, as given by Theorem IV.2. However, in order for $t = (N-K)/2$, the number of consecutive parity frequencies should be even. This creates no problem for an RS code, however, for a BCH code an even number of consecutive parity frequencies is not always possible.

If N is odd, then there is no problem in constructing a t-error correcting BCH code with $2t$ consecutive parity frequencies. Such a code will be of the form $(N, N-2t)$. If K is even, and N is odd, then ω^0 must be included as another parity frequency. For N even, either $\omega^0 = 1$ or $\omega^{\frac{N}{2}} = -1$ or both must be included as parity frequencies. As an example, Figure IV.2 depicts the parity frequency locations for the different possible combination of N and K given that $t = 2$, and $\omega = e^{j\frac{2\pi}{N}}$. Note that other primitive roots can used; however, for simplicity, they will not be considered. The properties of the code are not changed.

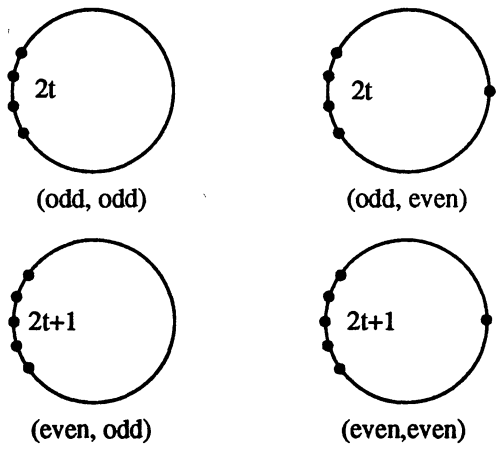


Figure IV.2: Parity Frequency Locations For t = 2 BCH Code.

An extra non-consecutive parity frequency increases the minimum distance of the code, and thus allows the code to detect more errors; however, the decoding algorithm for RS and BCH codes relies on the fact that the parity frequencies are consecutive. Thus, it will be the number of consecutive parity frequencies that will be used to determine the number of errors that a BCH code can correct. Henceforth, all BCH codes will have odd N and odd K, so that $2t = N - K$.

Decoding BCH and RS Codes

In Chapter III, it was seen that the main decoding problem is to determine the error locations. Once these are known, either of the two previous methods discussed in Chapter III for finding the correct information word can be used. Recall, one method used the syndrome equations to find the error magnitudes, which allowed the received word to be corrected. The second method merely deleted equations from the system of encoding equations and estimated the information word directly.

Finding the error locations could be accomplished by a direct search. Such a method would search for the correct $L = \{l_1, \ldots, l_\mu\}$ were μ is the number of errors, such that the syndrome, $s \in span(h_{l_1}, \ldots, h_{l_\mu})$. Clearly, a better method would be desirable.

Due to the structure of the DFT matrix and the choice of using consecutive parity frequencies, such a method exists. The method is valid for both BCH and RS codes. It consists of the following five steps:

1) Compute the syndromes using (III.4). If the code is a BCH code, i.e. $C \subset \mathbf{R}^N$, then the complex parity check matrix should be used.
2) Determine the number of errors.
3) Solve for the error locator polynomial.
4) Find the roots of the error locator polynomial; the roots give the error locations.

5) Solve for the information word using either of the two previously mentioned methods.

In the error correction literature, this process is known as the Peterson-Gorenstein-Zieler decoder. However, since the decoding takes place in the complex field, determining the error locations can be viewed as finding the frequencies of complex exponentials. In the signal processing literature, steps (2)-(4) are known as *Prony's method*, [Mpl87].

Assume that an (N,K) RS code has parity frequencies at positions $K,\ldots,N-1$. Then equation (III.4) has the form,

$$\begin{bmatrix} s_0 \\ s_1 \\ \cdot \\ \cdot \\ \cdot \\ s_{N-K-1} \end{bmatrix} = \begin{bmatrix} 1 & \omega^K & \cdots & \omega^{K(N-1)} \\ 1 & \omega^{K+1} & \cdots & \omega^{(K+1)(N-1)} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 1 & \omega^{N-1} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} e_0 \\ e_1 \\ \cdot \\ \cdot \\ \cdot \\ e_{N-1} \end{bmatrix}, \qquad (IV.5)$$

where $\omega = \exp(j2\pi/N)$. If there are μ errors, then the error vector, e, will be non-zero only at those positions given by $L = \{l_1,\ldots,l_\mu\} \subset [0,\ldots,N-1]$. Explicitly, the syndromes are given by

$$\begin{aligned} s_m &= \sum_{i=1}^{\mu} \omega^{(m+K)l_i} e_{l_i}, \qquad m = 0,\ldots,N-K-1, \\ &= \sum_{m=1}^{\mu} e_{l_i} \exp\left[j\frac{2\pi}{N}(l_i)(m+K) \right]. \end{aligned} \qquad (IV.6)$$

Equation (IV.6) depicts how each syndrome is of the form $\sum_{i=1}^{\mu} A_i \exp[j\omega_i(t+t_0)]$. Thus, each syndrome is a summation of complex sinusoids, where t is discrete and t_0 known. From the $N-K-1$ samples, μ frequencies must be determined. These equations are nonlinear with respect to the $\{l_i\}$. Prony's method provides a means for solving these nonlinear equations.

The first part of Prony's method is to determine the number of errors; or alternately, to determine the number of sinusoids. Again, suppose that there are μ errors at locations, $l_1,\ldots,l_\mu$ with values $e_{l_1},\ldots,e_{l_\mu}$. Form the following matrix of syndromes,

$$A=\begin{bmatrix} S_0 & S_1 & \cdots & S_{t-1} \\ S_1 & S_2 & \cdots & S_t \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ S_{t-1} & S_t & \cdots & S_{2t-2} \end{bmatrix}, \qquad \text{or}$$

$$A=[a_{ij}]=S_{i+j-2}, \qquad i,j=1,\ldots,t \tag{IV.7}$$

It is assumed that $N-K=2t$. Using this matrix, the number of errors can be determined by using the following theorem:

THEOREM IV.4: The rank of the syndrome matrix, A, is equal to the number of errors.

PROOF: Decompose A into the form $A=MBM^T$ where B is diagonal, and M is of full rank. First, let $\mu \le t$ and let $\{e_{l_1},\ldots,e_{l_t}\}$ be the set of error amplitudes with $e_{l_i}=0$ if $i>\mu$. The index positions $l_{\mu+1},\ldots l_t$, corresponding to zero amplitude error positions, can be chosen arbitrarily, however all $l_1,\ldots,l_t$ must be distinct.

Now consider the Vandermonde matrix,

$$M=\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \omega^{l_1} & \omega^{l_2} & \cdots & \omega^{l_t} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \omega^{(t-1)l_1} & \omega^{(t-1)l_2} & \cdots & \omega^{(t-1)l_t} \end{bmatrix},$$

and the diagonal matrix,

$$B = \begin{bmatrix} e_{l_1}\omega^{Kl_1} & 0 & \cdots & 0 \\ 0 & e_{l_2}\omega^{Kl_2} & \cdots & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & \cdots & e_{l_t}\omega^{Kl_t} \end{bmatrix}.$$

Now consider the product MBM^T. Since $m_{ik} = \omega^{(i-1)l_k}$ and $b_{kn} = e_{l_n}\omega^{Kl_n}\delta_{kn}$, where

$$\delta_{kn} = \begin{cases} 1, & k = n \\ 0, & otherwise \end{cases}$$

then,

$$\begin{aligned} (MBM^T)_{ij} &= \sum_{n=1}^{t}\sum_{k=1}^{t} \omega^{(i-1)l_k}\omega^{(j-1)l_n} e_{l_k}\omega^{Kl_n}\delta_{kn} \\ &= \sum_{n=1}^{t} \omega^{(i-1)l_n}\omega^{(j-1)l_n} e_{l_n}\omega^{Kl_n} \\ &= \sum_{n=1}^{t} e_{l_n}\omega^{l_n(K+i+j-2)} \\ &= \sum_{n=1}^{\mu} e_{l_n}\omega^{l_n(K+i+j-2)} \\ &= S_{i+j-2} \\ &= A_{ij}. \end{aligned}$$

Therefore, $A = MBM^T$. Since M is a Vandermonde matrix with distinct $\omega^{l_1}, \ldots, \omega^{l_t}$, it is nonsingular and of rank t. The matrix B, however, is of rank μ. Thus,

$$rank(A) = rank(MBM^T) = \mu,$$

the number of errors.

■

By determining the rank of the syndrome matrix, A, the number of errors can be found. The reader familiar with spectral estimation techniques will recognize that the syndrome matrix has the same form as a linear prediction matrix that arises in autoregressive (AR) spectral modelling. In the third step of Prony's method, finding the error locator polynomial is the same as fitting a μ^{th} order AR model to the syndromes.

Consider the following polynomial:

$$\alpha(z) = \left(z - \omega^{l_1}\right)\cdots\left(z - \omega^{l_\mu}\right) \qquad (IV.8)$$
$$= z^{\mu} + \alpha_1 z^{\mu-1} + \cdots + \alpha_{\mu-1} z + \alpha_{\mu}.$$

The roots of $\alpha(z)$ give the error locations, and so it is called the *error locator polynomial.* It will be shown that the syndromes can be used to find the coefficients of the error locator polynomial.

For $\mu \leq t$ errors, the following syndromes are known:

$$s_0 = e_{l_1}\omega^{Kl_1} + \cdots + e_{l_\mu}\omega^{Kl_\mu}$$
$$s_1 = e_{l_1}\omega^{(K+1)l_1} + \cdots + e_{l_\mu}\omega^{(K+1)l_\mu}$$
$$\vdots \qquad \vdots$$
$$s_{2t-1} = e_{l_1}\omega^{(K+2t-1)l_1} + \cdots + e_{l_\mu}\omega^{(K+2t-1)l_\mu}$$

Using the first $\mu + 1$ syndromes, a relationship between the syndromes and the coefficients of the error locator polynomial will be derived. First multiply $s_0, \ldots, s_{\mu-1}, s_\mu$ by $\alpha_\mu, \ldots, \alpha_1, 1$, respectively, giving

$$\alpha_\mu s_0 = \alpha_\mu e_{l_1}\omega^{Kl_1} + \cdots + \alpha_\mu e_{l_\mu}\omega^{Kl_\mu}$$

$$\vdots$$

$$\alpha_1 s_{\mu-1} = \alpha_1 e_{l_1}\omega^{(K+\mu-1)l_1} + \cdots + \alpha_1 e_{l_\mu}\omega^{(K+\mu-1)l_\mu}$$

$$s_\mu = e_{l_1}\omega^{(K+\mu)l_1} + \cdots + e_{l_\mu}\omega^{(K+\mu)l_\mu}.$$

Second, sum these equations. Rearranging terms, and using (IV.8),

$$\alpha_\mu s_0 + \cdots + \alpha_1 s_{\mu-1} + s_\mu = e_{l_1}\omega^{Kl_1}\left(\alpha_\mu + \cdots + \alpha_1\left(\omega^{l_1}\right)^{(\mu-1)} + \left(\omega^{l_1}\right)^{\mu}\right) +$$

$$\vdots$$

$$+ e_{l_\mu}\omega^{Kl_\mu}\left(\alpha_\mu + \cdots + \alpha_1\left(\omega^{l_\mu}\right)^{(\mu-1)} + \left(\omega^{l_\mu}\right)^{\mu}\right)$$

$$= \sum_{i=1}^{\mu} e_{l_i}\omega^{Kl_i}\alpha\left(\omega^{l_i}\right) = 0,$$

since ω^{l_i} is a root of $\alpha(z)$ for $i = 1, \ldots, \mu$. Thus, we see that

$$[s_0, \ldots, s_{\mu-1}, s_\mu]\begin{bmatrix} \alpha_\mu \\ \vdots \\ \alpha_1 \\ 1 \end{bmatrix} = 0.$$

In the same manner, the following set of homogeneous equations can be formed:

$$\begin{bmatrix} s_0 & s_1 & \cdots & s_\mu \\ s_1 & s_2 & \cdots & s_{\mu+1} \\ \vdots & \vdots & & \vdots \\ s_{\mu-1} & s_\mu & \cdots & s_{2\mu-1} \end{bmatrix}\begin{bmatrix} \alpha_\mu \\ \alpha_{\mu-1} \\ \vdots \\ 1 \end{bmatrix} = 0. \qquad (IV.9)$$

To solve for the error locator coefficients, we rearrange (IV.9) and solve

$$-\begin{bmatrix} s_{\mu} \\ s_{\mu+1} \\ \cdot \\ \cdot \\ \cdot \\ s_{2\mu-1} \end{bmatrix} = \begin{bmatrix} s_0 & s_1 & \cdots & s_{\mu-1} \\ s_1 & s_2 & \cdots & s_{\mu} \\ \cdot & & \cdot & \\ \cdot & & \cdot & \\ \cdot & & \cdot & \\ s_{\mu-1} & s_{\mu} & \cdots & s_{2\mu-2} \end{bmatrix} \begin{bmatrix} \alpha_{\mu} \\ \alpha_{\mu-1} \\ \cdot \\ \cdot \\ \cdot \\ \alpha_1 \end{bmatrix}. \qquad (IV.10)$$

Thus, by solving the system of μ equations specified in (IV.10), the error locator polynomial can be found. Theorem IV.4 guarantees that there is a unique solution to (IV.10).

Once the error locator polynomial has been found, the locations are given by the roots of $\alpha(z)$. The roots are commonly found by searching over the N powers of ω. Because the decoding procedure is performed in the complex field, BCH codes over the real field will still need a complex parity check matrix. After the error locations are known, the decoding process is essentially complete. Either of the two previously mentioned methods can be used to get the information word.

CHAPTER V

REAL NUMBER BCH AND REED-SOLOMON CODES IN ADDITIVE NOISE

The decoding procedures for real number BCH and RS codes presented in Chapter IV work perfectly given that the real number codewords are known precisely. However, from Chapter II, it was seen that the transmission of real and complex numbers required a data compression code. The most common a straight forward compression method is quantization.

When each element in the codeword is quantized, noise is introduced. In general, the quantization noise is a function of many variables, including the number of quantization levels and the statistical distribution of the input data. This noise affects the decoding procedure, creating the possibility that the decoder will fail to find the correct error locations even though the number of errors is less than or equal to t. Recall, t is the maximum number of errors that the error correction code can correct.

Consider the two different real number transmission systems depicted in Figure V.1. Suppose that both sources are identical and that they output K real numbers at a time. The first system quantizes the real numbered source data before applying an error correction code. Since the channel coder data is digital, a finite field code is appropriate. The second system uses a real number code directly on the source output and then quantizes the channel coder output.

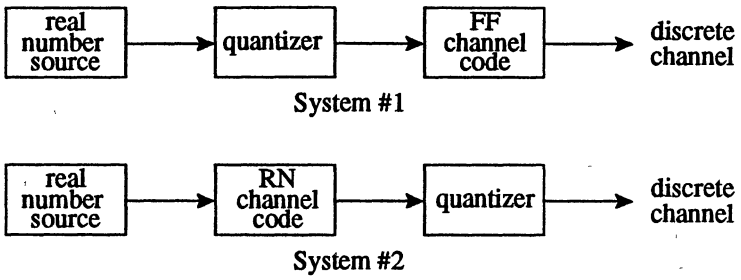


Figure V.1: Two Real Number Data Transmission Systems

Assume that both channel coders in Figure V.1 have the same code parameters, i.e. N, K and t. Also, assume that both quantizers discretize the real numbered data to b bits and that a t-error correcting (N,K) finite field code based upon $\mathbf{F}_{2^b}$ exists. With these assumptions, the discrete channel data rates for the two cases are equal and are given by

$$\text{Channel Data Rate } = \frac{Nb \text{ bits}}{K \text{ source symbols}}.$$

The receivers for these two systems, (henceforth referred to as system #1 and system #2), will decode the received vectors and arrive at an estimate of the original data vector. Denote this estimate by $\hat{d}$. One possible way to compare the two systems, would be to compute the total mean squared error, given by

$$MSE = \frac{1}{K} E\{(d-\hat{d})^T (d-\hat{d})\}. \qquad (V.1)$$

In general, comparing the two systems requires that the distortion between the true data vector and the estimated data vectors of system #1 and system #2 can be measured. The MSE is a convenient distortion measure since it is mathematically tractable. More

general distortion measures exist, ([Gra80] gives some examples for speech), however, these measures are usually source dependent. It is common to refer to the MSE as the *quadratic distortion measure.*

Let $d \in \mathbf{R}^K$. For convenience, assume that the vector d is independent, identical, and normally distributed with zero mean and unit variance, i.e. $d_i \sim N(0,1)$ for $i = 0,\ldots,K-1$ with $E\{d_i d_j\} = 0$ if $i \neq j$. (In short, $E\{d_i d_j\} = (R_d)_{ij} = \delta_{ij}$ or $R_d = I_K$.) In the first system, the quantizer introduces a small amount of error to the information word. Thus, let

$$Q[d] = d + q_d, \tag{V.2}$$

where $Q[\cdot]$ is the quantization operation and q_d is the quantization "noise" vector that is added to the data vector. Assuming that $Q[\cdot]$ is either a LM quantizer or a uniform quantizer optimized for normally distributed data, then $Q[d]$ is unbiased, i.e. $E\{Q[d]\} = d$.

For the first system, let $\mathrm{M1_{FF}}$ be the mean squared error given that the number of transmission errors, μ, does not exceed the error correction limits of the code, t. In this case, $Q[d]$ is the estimate obtained by the channel decoder. Thus, $\hat{d} = Q[d]$, and $\mathrm{M1_{FF}}$ is fixed by the quantizer.

If the number of errors exceeds the limits of the code, then an incorrect estimate of the data word will be obtained, and it is likely that the MSE will be high. Let $\mathrm{M2_{FF}}$ denote the average of this value. If P_E is the probability of an error in the decoded information vector, i.e. $P_E = prob(\text{Number of Errors} > t)$, then the total MSE for the finite field case is given by,

$$MSE_{FF} = (1 - P_E)M1_{FF} + P_E M2_{FF}. \tag{V.3}$$

If P_E is sufficiently small compared to the ratio $M1_{FF}/M2_{FF}$, then $MSE_{FF} \cong M1_{FF}$.

In the second system, the real valued codeword is quantized. Thus,

$$Q[c] = c + q, \tag{V.4}$$

where q is the $N \times 1$ quantization noise vector added to the codeword. For the real number case, the channel decoder will also determine an estimate of the information word, $\hat{d}$. The mean squared error for this case is again given by (V.1).

Similar to the finite field case, write

$$MSE_{RN} = (1 - P_E)M1_{RN} + P_E M2_{RN}, \qquad (V.5)$$

where P_E is the probability of a decoding error. In the finite field system, the probability of a decoding error was equal to the probability that more than t errors occurred during the transmission of the N symbols. In the real number system, the probability of an error is greater since the decoder can fail even when fewer than t errors are present.

For real number codes, it will be convenient to write the probability of a decoding error as

$$P_E = P_{UNC} + P_{DF}, \qquad (V.6)$$

where P_{UNC} is the probability of an uncorrectable error pattern and P_{DF} is the probability of a decoding failure. A *decoding failure* occurs when the decoder makes an error even though $\mu \leq t$. Note that $P_{UNC} = prob(\mu > t)$ is equal to P_E in the finite field system.

Let $M1_{RN}$ be the mean squared error given that the decoder does not fail and $\mu \leq t$. It is a function of the number of errors, the error locations, the code parameters and the quantizer. Let $M2_{RN}$ be the mean squared error given that either a decoding failure has occurred or that $\mu > t$. For a real number code, in order for the mean squared error to be approximately equal to $M1_{RN}$, P_{UNC} and P_{DF} must be sufficiently small compared to the ratio $M1_{RN}/M2_{RN}$.

In order to compare the two systems in Figure V.1, P_{DF} and $M1_{FF}$ for the real number code must be examined. The behavior of $M1_{RN}$ is a source coding property of the real number codes, while the behavior of P_{DF} is a channel coding property. The total MSE is

a joint source-channel property. Before proceeding with these properties of real number BCH and Reed-Solomon (RS) codes, a more detailed look at the generator and parity check matrices is presented.

More On G and H

From the previous chapter, it was seen that the generator and parity check matrices for RN BCH and Reed-Solomon codes can be formed from the columns of the DFT matrix. Since these matrices are not unique, the procedure in Chapter IV is not the only way to form G and H. However, by using the DFT, nice properties result.

These results will prove useful when looking at the source and channel coding properties of real number BCH and RS codes. Three primary results will be derived is this section. The first result is really a normalizing condition, and is fundamental. The second result pertains to the singular values of G, which are instrumental in determining the accuracy of the estimated data word. Finally, the third result gives a unitary relationship between the real and complex versions of both the generator and parity check matrices for BCH codes. After this final result is presented, the focus of this document will be on BCH codes, since real vector spaces are easier to visualize than complex vector spaces.

Normalizing Conditions

Consider a Reed-Solomon code based on the columns of the DFT matrix, like those presented in Chapter IV. Since G and H can be multiplied by scalars without changing the properties of the code, it follows that for

$$W_N = [w_0, \ldots, w_{K-1}, w_K, \ldots, w_{N-1}],$$

the generator and parity check matrices for a RS code can be written as

$$G = \sqrt{\left(\frac{N}{K}\right)}[w_0, \ldots, w_{K-1}], \tag{V.7}$$

and

$$H = \sqrt{\left(\frac{N}{N-K}\right)}[w_K, \ldots, w_{N-1}]. \tag{V.8}$$

(This is the same construction as given by Definition IV.4.) With these matrices, the following theorem holds:

THEOREM V.1: The generator and parity check matrices for an (N,K) Reed-Solomon code given by (V.7) and (V.8), respectively, have the following properties:

(1a) $G^T G = \frac{N}{K} I_K$

(1b) $(GG^T)_{ii} = 1$ for $i = 1, \ldots, N$

(2a) $H^T H = \frac{N}{N-K} I_{N-K}$

(2b) $(HH^T)_{ii} = 1$ for $i = 1, \ldots, N$

PROOF: One only needs to prove parts (1a) and (1b), since the proof for (1a) and (1b) is the same as the proof for parts (2a) and (2b). (1a) follows directly from (V.7) and the fact that the DFT matrix is unitary. To prove (1b), consider that from (V.7), it follows that

$$(GG^T)_{ii} = \frac{N}{K} \sum_{k=0}^{K-1} \frac{(\omega^{ik})}{\sqrt{N}} \frac{(\omega^{ik})^*}{\sqrt{N}}$$

$$= 1,$$

where $\omega = e^{-j\frac{2\pi}{N}}$.

■

Singular Values

Using Theorem V.1, the singular values of the generator and parity check matrices are apparent. This result is given in Theorem V.2. The reader not familiar with the singular value decomposition can consult Appendix B.

THEOREM V.2: Let an RS code with generator and parity check matrices specified by (V.7) and (V.8) respectively, be given. Then the singular values of G are equal to $\sqrt{N/K}$, and the singular values of H are equal to $\sqrt{N/(N-K)}$.

PROOF: First, for a given matrix A, the matrix A^TA is positive semi-definite. Thus, since the singular values of A are the positive square roots of the eigenvalues of A^TA, (1a) and (2a) of Theorem V.1 give the desired result.

■

In the next section, it will be shown that when there are no errors, $M1_{RN}$ is related to the singular values of the generator matrix. However, when there are errors present, the second method of estimating the information word (given in Chapter III) requires that the rows of G corresponding to the error locations must be deleted. In this case, the $M1_{RN}$ depends upon the singular values of this deleted matrix.

If the error locations are denoted by the index set $L = \{l_1, \ldots, l_\mu\}$, then let

$$L^c = [0, \ldots, N-1] - L,$$

where the subtraction operation is between sets. L^c will be called the *complement of* L. (L will be called an *error index set.*) Let

$$G = \begin{bmatrix} g_0^T \\ g_1^T \\ . \\ . \\ g_{N-1}^T \end{bmatrix}. \qquad (V.9)$$

Then G_{L^c} is the matrix formed by keeping the rows of G whose indices are specified by L^c. Equivalently, G_{L^c} is the matrix formed by deleting those rows specified by L. Some relationships regarding the singular values of G_{L^c} are given by the following theorem:

THEOREM V.3: Given an RS code as in Theorem V.2 and an error index set L with $|L| = \mu \leq t$, let $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_K$ be the singular values of G_{L^c}. Then the following are true:

(1) $\sigma_K > 0$

(2) $\sum_{i=1}^{K} \sigma_i^2 = N - |L|$

(3) If $|L| = 1$, then $\sigma_1^2 = \cdots = \sigma_{K-1}^2 = \frac{N}{K}$, $\sigma_K^2 = \frac{N}{K} - 1$

(4) If $|L| = 2$, then

$$\sigma_1^2 = \cdots = \sigma_{K-2}^2 = \frac{N}{K}$$

$$\sigma_{K-1}^2 = \frac{N}{K} - 1 + \gamma, \quad \sigma_K^2 = \frac{N}{K} - 1 - \gamma$$

where $\gamma = |g_{l_1}^T g_{l_2}|$

(5) In general, if $|L| = \mu$, then

$$\sigma_1^2 = \cdots = \sigma_{K-\mu}^2 = \frac{N}{K}$$

$$\sigma_{K-i+1}^2 = \frac{N}{K} - \lambda_i \quad i = 1, \ldots, \mu$$

where $\lambda_1 \geq \cdots \geq \lambda_\mu > 0$ are the real eigenvalues of the matrix

$$\begin{bmatrix} g_{l_1}^T g_{l_1} & g_{l_1}^T g_{l_2} & \cdots & g_{l_1}^T g_{l_\mu} \\ g_{l_2}^T g_{l_1} & g_{l_2}^T g_{l_2} & \cdots & g_{l_2}^T g_{l_\mu} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ g_{l_\mu}^T g_{l_1} & g_{l_\mu}^T g_{l_2} & \cdots & g_{l_\mu}^T g_{l_\mu} \end{bmatrix}. \qquad (V.10)$$

PROOF: (1) is true if and only if $rank(G_{L^c}) = K$. Theorem III.7 guarantees that this is the case.

(2) Consider that $trace(GG^T) = N$ from Theorem V.1. Deleting $|L|$ rows from G decreases the size of $G_{L^c}G_{L^c}^T$ to $(N - |L|) \times (N - |L|)$. However, the diagonal elements are still one, thus $trace\left(G_{L^c}G_{L^c}^T\right) = N - |L|$. Since the trace of a square matrix equals the sum of it eigenvalues, (2) follows.

(3) and (4) are special cases of (5). Assuming (5), if $|L| = 1$, then the matrix in (V.10) has an eigenvalue of one. Thus, (3) holds. If $|L| = 2$, then (V.10) has the form

$$\begin{bmatrix} 1 & g_{l_1}^T g_{l_2} \\ g_{l_2}^T g_{l_1} & 1 \end{bmatrix}.$$

This matrix is hermitian (symmetric for BCH) and has real eigenvalues equal to $1 \pm |g_{l_1}^T g_{l_2}|$, where $|g_{l_1}^T g_{l_2}| < 1$. Thus, (4) is true.

Now show item (5). Expanding $G^T G$ as

$$G^T G = \sum_{m=0}^{N-1} g_m g_m^T = \frac{N}{K} I_K.$$

Now expanding the deleted matrix in terms of outer products gives

$$G_{L^c} G_{L^c}^T = \frac{N}{K} I_K - \sum_{m=1}^{\mu} g_{l_m} g_{l_m}^T.$$

By using Theorem B.4 and Theorem III.7, it is known that $K - \mu$ eigenvalues of $G_{L^c} G_{L^c}^T$ will have magnitudes equal to N/K, while the remaining μ non-zero eigenvalues, denoted by λ_i, $i = 1, \ldots, \mu$, will have magnitudes less than or equal to N/K. Let v_i be the corresponding eigenvectors. Write

$$\left(G_{L^c} G_{L^c}^T\right) v_i = \lambda v_i$$

$$\left(\frac{N}{K} I_K - \sum_{m=1}^{\mu} g_{l_m} g_{l_m}^T\right) v_i = \lambda v_i$$

$$\sum_{m=1}^{\mu} g_{l_m} g_{l_m}^T v_i = \left(\frac{N}{K} - \lambda\right) v_i.$$

The matrix $\sum_{m=1}^{\mu} g_{l_m} g_{l_m}^T$ is of rank μ. Let the μ nonzero eigenvalues of $\sum_{m=1}^{\mu} g_{l_m} g_{l_m}^T$ be given by $\gamma_1, \ldots, \gamma_\mu$. Thus,

$$\left(\frac{N}{K} - \lambda_i\right) v_i = \gamma_i v_i \quad \text{for } i = 1, \ldots, \mu.$$

Rearranging, the eigenvalues of $G_{L^c} G_{L^c}^T$ can be written as

$$\lambda_i = \frac{N}{K} - \gamma_i \qquad i = 1, \ldots, \mu. \tag{V.11}$$

Now let

$$G' = \begin{bmatrix} g_{l_1}^T \\ \cdot \\ \cdot \\ g_{l_\mu}^T \end{bmatrix}$$

then $\sum_{m=1}^{\mu} g_{l_m} g_{l_m}^T = G'(G')^T$. The nonzero eigenvalues of $G'(G')^T$ and $(G')^T G'$ are the same. However, the matrix in (V.10) is the same as $(G')^T G'$, so (5) follows from (V.11).

■

Relationship Between G_R, H_R and G_C, H_C

Up to this point, most of the results have been presented using RS codes. This means that the generator and parity check matrices are complex. Since BCH codes are Reed-Solomon codes with the additional conjugate constraint, the previous normalization and singular value results hold. However, as demonstrated in Chapter IV, for BCH codes, it is possible to obtain real generator and parity check matrices from the DFT based complex matrices. The purpose of this section is to show that real G and H exist which have the same properties as those given in theorems V.1 - V.3.

Consider the (5,3) example in Chapter IV. Let G_C be the complex BCH generator matrix given by

$$G_C = [w_0, w_1, w_1^*].$$

It was seen that a real generator matrix, G_R, was given by

$$G_R = \left[w_0, \frac{w_1 + w_1^*}{\sqrt{2}}, -j\left(\frac{w_1 - w_1^*}{\sqrt{2}} \right) \right].$$

Alternately, write G_R as

$$G_{\mathbf{R}} = [w_0, w_1, w_1^*] \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{-j}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & \frac{j}{\sqrt{2}} \end{bmatrix}$$

$$= G_{\mathbf{C}} U_G.$$

Note that U_G is a unitary matrix.

Extending this example to the general case for the generator matrix is not difficult. Also, the real and complex parity check matrices are related by another unitary matrix of a similar form. Since multiplication by a constant does not affect the unitary relationship, the following theorem is true.

THEOREM V.4: Given a BCH code with $G_{\mathbf{C}}$ and $H_{\mathbf{C}}$ normalized as in Theorem V.1, then there exists real matrices $G_{\mathbf{R}}$ and $H_{\mathbf{R}}$ for which theorems V.1, V.2, and V.3 hold.

PROOF: Find unitary matrices U_G and U_H such that

$$G_{\mathbf{R}} = G_{\mathbf{C}} U_G \qquad (V.12)$$

$$H_{\mathbf{R}} = H_{\mathbf{C}} U_H. \qquad (V.13)$$

Theorem V.1 is true since U_G and U_H are unitary. Theorems V.2 and V.3 follow from Theorem V.1 in the same way as before.

■

Since real codespaces and syndrome spaces are easier to visualize, primarily BCH codes with real generator and parity check matrices will be used to gain insight into the source and especially the channel coding properties of RN BCH and RS codes. These

properties rely upon the singular values of the generator and parity check matrices. Since the unitary relationship does not affect the singular values, (V.12) and (V.13) ensure that these insights are not restricted to only the real versions of the codes.

Source Coding Properties

In this section, $M1_{RN}$ is examined. Recall, that $M1_{RN}$ is the expected value of the mean squared error between the actual information word and the estimated data word given that the correct error locations have been found. If the probability of an error, as given by (V.6), is sufficiently small, then the total mean squared error for the system is approximately equal to $M1_{RN}$.

In Chapter III, two different methods for determining an estimate of the information word were given. The first method used the parity check matrix, while the second used only the generator matrix. In the finite field case or the real/complex field case with infinite precision, both methods are equivalent. However, with quantization noise, the received vector becomes

$$r = c + q + e, \tag{V.14}$$

where q is the quantization noise vector, and e is the transmission error vector. It is no longer clear that the two methods will give the same estimate. But before deriving expressions for these two methods, a bit of notation is needed.

DEFINITION V.5: For a given index set J, the *selection matrix*, S_J, is a $N \times |J|$ matrix created by selecting the columns of I_N that correspond to the indices in J.

If $J = \{j_1, \ldots, j_n\}$, then let the first column of S_J equal column number j_1 of I_N. Thus, since J is an ordered set, S_J is unique. Using a selection matrix, the deleted generator matrix, G_{L^c} can be written as

$$G_{L^c} = S_{L^c}^T G. \tag{V.15}$$

In addition, the "sparse" $N \times 1$ transmission error vector e, can be written as

$$e = S_L \begin{bmatrix} e_{l_1} \\ \cdot \\ \cdot \\ e_{l_\mu} \end{bmatrix}. \tag{V.16}$$

The following properties concerning selection matrices are true and are easily verified:

1) $S_J^T S_J = I_{|J|}$

2) $S_J S_J^T + S_{J^c} S_{J^c}^T = I_N$

Now assume that L is the error index set with $|L| = \mu$ and that L is known. Using the parity check equation, an estimate of the error vector is given by

$$\begin{aligned} \hat{e} &= S_L (H_L^T)^+ s \\ &= S_L (H^T S_L)^+ H^T r, \end{aligned} \tag{V.17}$$

where + denotes the pseudo-inverse of a matrix.

By subtracting $\hat{e}$ from r, an estimate of the information word is given by

$$\begin{aligned} \hat{d} &= G^+ (r - \hat{e}) \\ &= G^+ \left(r - S_L (H^T S_L)^+ H^T r \right) \\ &= G^+ \left(I_N - S_L (H^T S_L)^+ H^T \right) r \end{aligned} \tag{V.18}$$

Thus, (V.18) gives the transformation from r to $\hat{d}$ for the first method of decoding.

For the second method, the equations designated by L in the encoding equation are deleted. Therefore, the estimate is given by

$$\hat{d} = \left(S_{L^c}^T G\right)^+ S_{L^c}^T r \qquad (V.19)$$

Experimentally, the two estimators given by (V.18) and (V.19) are identical, and their equality is left as a conjecture.

CONJECTURE V.6: The two methods for estimating the information word as specified by (V.18) and (V.19) are identical, i.e.

$$\left(S_{L^c}^T G\right)^+ S_{L^c}^T = G^+\left(I_N - S_L (H^T S_L)^+ H^T\right). \qquad (V.20)$$

Note that the two linear operators on r given by (V.18) and (V.19) are independent of the quantization noise, q. It should be clear that the two estimators are equal if $q = 0$, since this case is merely the infinite precision case. In order for the two estimators to be equal for all q, then (V.20) must be true. By assuming this conjecture, only one estimation method needs to be analyzed.

M1 and the Singular Values of G_{L^c}

Again, for convenience, assume that the information vector is independent, normal and identically distributed, with $E\{d\} = 0$ and $R_d = E\{dd^T\} = I_K$. Although, this assumption is not always valid, it is common to model a data stream which has been source compacted in this fashion, [Jay84].

Given that the error pattern is correctable, the mean square error for the finite field system in Figure V.1 is determined by the quantizer. Assume that $E\{q_d\} = 0$ and $R_{q_d} = \sigma_q^2 I_K$. Then the mean squared error for the finite field case is given by

$$MSE_{FF} = \frac{1}{K} E\{(d-\hat{d})^T (d-\hat{d})\}$$

$$= \frac{1}{K} E\{q_d^T q_d\}$$

$$= \sigma_q^2. \qquad (V.21)$$

For the real number case, write (V.19) as

$$\hat{d} = (G_{L^c})^+ r_{L^c}$$

$$= (G_{L^c})^+ (c_{L^c} + q_{L^c})$$

$$= (G_{L^c})^+ q_{L^c} + d \qquad (V.22)$$

The final result is due to the fact that any K or more equations from $c = Gd$ will give d. Since $R_d = I_K$,

$$R_c = GR_dG^T$$

$$= GG^T. \qquad (V.23)$$

Using Theorem V.1b, the variance of c_i is unity for $i = 1,\ldots,N$. In addition, c is normally distributed. Thus, the same quantizer can be used in the real case as was used in the finite field case.

Using (V.22), the following theorem gives M1 for the real number case.

THEOREM V.7: Given an (N,K) real number RS code and error index set L, with each $d_i \sim N(0,1)$, independent, and identically distributed and q zero mean with $R_q = \sigma_q^2 I_N$, then the mean squared error of the data estimate is given by

$$M1_{RN} = \frac{\sigma_q^2}{K} \sum_{i=1}^{K} \frac{1}{\sigma_i^2},$$

where $\sigma_1, \ldots, \sigma_K$ are the singular values of G_{L^c}.

PROOF: First, let the SVD of $G_{L^c} = U\Sigma V^T$. Then

$$MI_{RN} = \frac{1}{K} E\{(d-\hat{d})^T (d-\hat{d})\}$$

$$= \frac{1}{K} E\left\{ q_{L^c}^T \left(G_{L^c}^+\right)^T G_{L^c}^+ q_{L^c} \right\}$$

$$= \frac{1}{K} E\left\{ q_{L^c}^T U^T (\Sigma^+)^T V V^T \Sigma^+ U q_{L^c} \right\}$$

$$= \frac{1}{K} E\left\{ q_{L^c}^T U^T \Lambda U q_{L^c} \right\}$$

where

$$\Lambda = \begin{bmatrix} S^{-1} & 0 \\ 0 & 0 \end{bmatrix} \text{ and}$$

$$S^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2} & \cdot & \cdot & \cdot & 0 \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ 0 & \cdot & \cdot & \cdot & \frac{1}{\sigma_K^2} \end{bmatrix}.$$

Now let $q' = U q_{L^c}$, so $E\{q'\} = 0$ and $R_{q'} = U R_{q_{L^c}} U^T = \sigma_q^2 I_{|L^c|}$. Thus,

$$MI_{RN} = \frac{1}{K} E\{(q')^T \Lambda q'\}$$

$$= \frac{\sigma_q^2}{K} \sum_{i=1}^{K} \frac{1}{\sigma_i^2}.$$

■

In the case where there are no errors, using Theorem V.2, $\sigma_1 = \ldots = \sigma_K = N/K$ so

$$M1_{RN} = \frac{K}{N}\sigma_q^2$$

$$= \frac{K}{N} M1_{FF}. \qquad (V.24)$$

Equation (V.24) implies that without a decoding error, the real number code in system #2 can achieve a lower MSE than what would be obtained with the finite field code in system #1. The extra redundancy added by the real number channel coder can be used to lower the MSE. However, the price paid for this benefit is that the channel decoding is not precise as it is for finite field channel codes. For a RN error correction code, $P_{DF} \neq 0$.

In general, $M1_{RN}$ depends upon the error locations, since the singular values in Theorem V.7 depend upon L. (Recall Theorem V.3.) In order to obtain an average mean squared error, one must average over all possible L. In the last section of this chapter, titled "Joint Source-Channel Coding", this averaging is done for (15,7) and (19,11) BCH codes.

Weighted Codes

In the previous section, the MSE was calculated on a per frame basis. Instead, $R_{d-\hat{d}}$ can be calculated. Assuming no errors and the same statistics of d and q as before,

$$E\{(d-\hat{d})(d-\hat{d})^T\} = G^+ q q^T (G^+)^T$$

$$= \sigma_q^2 G^+ (G^+)^T.$$

Now using $G^+ = (G^T G)^{-1} G^T = \frac{K}{N} G^T$,

$$E\{(d-\hat{d})(d-\hat{d})^T\} = \sigma_q^2 \frac{K^2}{N^2} G^T G$$

$$= \frac{K}{N}\sigma_q^2 I_K. \qquad (V.25)$$

This is the expected result.

Equation (V.25) shows that with no errors, each element of the information word is estimated with the same precision. This is good assuming that each element of the information word is of equal importance. However, this is not always the case.

For example, certain source coding methods, like transform coding methods, give a vector of data elements that are not equally important. Many predictive coding techniques also produce a vector of elements which are of unequal importance. For example, the reflection coefficients transmitted in many voice coders are usually coded with varying degrees of accuracy, [Tre82].

Let $d^T = [d_1, \ldots, d_K]$ be a zero mean information vector. Assume that d is independent and that each d_i has been scaled so that $E\{dd^T\} = R_d = I_K$. In the finite field case, each element of d would be quantized with a different number of bits depending on the importance of that element's accuracy. For a real number code, a different approach is taken.

Since the codeword is quantized in the RN case, allocating a varying number of bits is not appropriate. It is desirable for c to be zero mean with each element having a variance of one. This condition allows for even quantization of the codeword, and a quantization error covariance matrix of the form $\sigma_q^2 I_N$. Thus, what is desired is a new generator matrix $\tilde{G}$ such that (1b) of Theorem V.1 holds, but

$$E\{(d-\hat{d})(d-\hat{d})\} = \frac{K}{N}\sigma_q^2 \begin{bmatrix} \alpha_1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \alpha_2 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & \alpha_K \end{bmatrix}. \qquad (V.26)$$

Equation (V.26) implies that those elements of $\hat{d}$ with smaller weight, α_i, will be represented more precisely. Thus, the relative importance of each element will be reflected in the weights: the smaller the weight, the more important.

A code with an error covariance given by (V.26) will be called a *weighted code.* Theorem V.8 gives the construction for weighted codes.

THEOREM V.8: Given a RS code with generator matrix G and parity check matrix H; a weighted version of this code with $R_{d-\hat{d}}$ specified by (V.26) is defined by the parity check matrix H and a generator matrix $\tilde{G}$, where

$$\tilde{G} = GW \tag{V.27}$$

with

$$W = \begin{bmatrix} \frac{1}{\sqrt{\alpha_1}} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \frac{1}{\sqrt{\alpha_2}} & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & \frac{1}{\sqrt{\alpha_K}} \end{bmatrix}, \tag{V.28}$$

and $\sum_{i=1}^{K} \frac{1}{\alpha_i} = K$.

PROOF: First note that normalizing condition (1a) in Theorem V.1 becomes

$$\tilde{G}^T\tilde{G} = W^T G^T G W = \frac{N}{K} W^2.$$

Second, note that (1b) becomes

$$(\tilde{G}\tilde{G}^T)_{ii} = \sum_{m=1}^{K} \frac{|g_{im}|^2}{\alpha_m}.$$

Before normalization, each g_{im} was a root of unity. Thus, $|g_{i1}| = \cdots = |g_{iK}|$. Also, from the case where all the weights are one, then (1b) in Theorem V.1 implies that $|g_{im}|^2 = 1/K$. Therefore, the weights must be normalized such that $\sum_{i=1}^{K} \frac{1}{\alpha_i} = K$ such that

$$(\tilde{G}\tilde{G})_{ii} = \frac{1}{K}\sum_{i=1}^{K}\frac{1}{\alpha_i}$$

$$= \frac{K}{K} = 1$$

Since condition (1b) is the same for $\tilde{G}$ as it is for G, the quantization of the codeword will still be even. By repeating the steps leading to (V.25),

$$E\{(d-\hat{d})(d-\hat{d})^T\} = \frac{K}{N}\sigma_q^2 (W^{-1})^2.$$

This is the same result as specified by the theorem.

To verify that the parity check matrix remains unchanged, compute

$$\begin{aligned} s &= H^T r \\ &= H^T(c+q+e) \\ &= H^T(\tilde{G}d+q+e) \\ &= H^T(GWd+q+e) \\ &= H^T(q+e). \end{aligned}$$

This is the same result as before.

■

Weighting a code is purely a source coding procedure. The channel coding properties are determined by H, which remains unchanged.

Theorem V.8 applies to RS codes. For a BCH code, the procedure is the same, except for the following restriction: if w_i corresponds to the i^{th} column of G and w_j corre-

sponds to the j[th] column of G, with $w_i = w_j^*$, then α_i must equal α_j. An example of a (7,3) BCH weighted code is given in Chapter VII, along with a computer simulation of its source coding properties.

Channel Coding Properties

The goal of this section is to somehow quantify the probability of a decoding failure for a specified real number BCH or RS code. Recall, that since the codeword must be quantized, the received vector has the form,

$$r = c + q + e. \tag{V.29}$$

Thus, the syndrome becomes

$$s = H^T(q + e). \tag{V.30}$$

For simplicity, assume that the codespace is real, i.e. $C \subset \mathbf{R}^N$. Then for every $q \in \mathbf{R}^N$, one can uniquely write

$$q = q_C + q_{C^\perp}$$

where $q_C \in C$ and $q_{C^\perp} \in C^\perp$. Substituting into (V.30), gives

$$\begin{aligned} s &= H^T(q_C + q_{C^\perp} + e) \\ &= H^T(q_{C^\perp} + e) \\ &= \Delta s + \bar{s}. \end{aligned} \tag{V.31}$$

For a general t error correcting code, the syndrome error subspace corresponding to an error location set L is the $|L|$-dimensional subspace $Im(H_L^T)$, with $|L| \leq t$. If e is a weight $|L|$ error vector, then $\bar{s} \in Im(H_L^T)$. Since, in general, $q_{C^\perp}$ is of weight N, it is very likely that $\Delta s \notin Im(H_L^T)$. Thus, Δs can be thought of as a perturbation or noise which displaces s from the syndrome error subspace. Note that

$$\bar{s} = E\{s\}, \tag{V.32}$$

and

$$\begin{aligned} R_{s-\bar{s}} &= E\{(s-\bar{s})(s-\bar{s})^T\} \\ &= E\{\Delta s \Delta s^T\} \\ &= E\{H^T q q^T H\} \\ &= \sigma_q^2 \frac{N}{N-K} I_{N-K}. \end{aligned} \tag{V.33}$$

Recall, without any quantization, the problem facing the decoder is to find the syndrome error subspace that contains s. Knowing the syndrome error subspace is equivalent to knowing the error locations. With $q \neq 0$, if e is of weight t or less, then $\bar{s}$ could be used to exactly specify the correct error subspace. However, only s is known and more than likely it does not lie in any proper syndrome error subspace. The problem now facing the decoder is to find an error index set $\hat{L}$ based upon the observation of the "noisy" syndrome, s.

An obvious decoding method is to choose the syndrome subspace that is closest to the syndrome. The distance between the syndrome subspace and the syndrome will be the standard Euclidean distance. Let P_J be the orthogonal projection onto the syndrome error subspace corresponding to the index set J, with $|J| = \mu \leq t$; μ is assumed to be known. Choosing the closest syndrome subspace corresponds to choosing $\hat{L}$ to be the J that satisfies

$$\min_J \| s - P_J s \| . \tag{V.34}$$

This decoding rule will be called the *nearest subspace decoding* (NSD) rule, since it finds the closest syndrome error subspace to s. Since q is zero mean and independent, the NSD rule will minimize the probability of a decoding failure. Analyzing P_{DF} for the NSD rule is easily done for BCH codes with a two dimensional syndrome space. For that reason, single error correcting BCH codes are investigated first.

Single Error Correcting Codes

For a given BCH code with N odd and $N-K=2$, let a single error exist with $L=\{0\}$. The syndrome space, depicted in Figure V.2, is two dimensional and it contains N one dimensional syndrome error subspaces. (Note that it is assumed that the BCH code was constructed in the fashion illustrated in Chapter IV. This construction insures the geometry of the subspaces depicted in Figure V.2. That is, the nearest syndrome error subspaces to $Im(H^T_{\{0\}})$ are $Im(H^T_{\{1\}})$ and $Im(H^T_{\{N-1\}})$

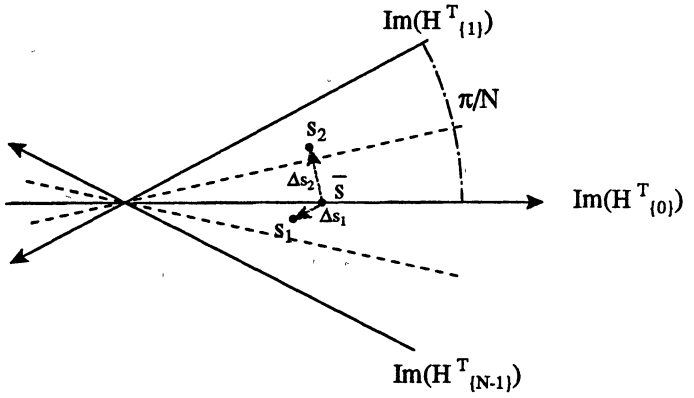


Figure V.2: Syndrome Space For (N,N-2) Code.

From the construction of the real parity check matrix given in Theorem V.4, it is not difficult to see that the syndrome error subspaces are equally spaced with the minimum angle between any two subspaces being equal to π/N. Earlier, Figure III.7 depicted the complete syndrome space for a (7,5) BCH code.

Since $L=\{0\}$, then it follows that $\bar{s} \in Im(H^T_{\{0\}})$. The pie shaped region around $Im(H^T_{\{0\}})$ bounded by the two dashed lines indicates the area in which s will be decoded

correctly. If s lies outside this region, then the decoder will fail. For example, the syndrome labeled s_1 in Figure V.2 will be decoded correctly since it is closest to $Im(H_{\{0\}}^T)$, while s_2 will result in a failure.

Since q is independent and identically distributed, by the Central Limit Theorem, s will approach a normal distribution as the block length increases. Equations (V.32) and (V.33) give the mean and covariance of s, respectively. Clearly, as the variance of s decreases, the probability of a decoding failure decreases. For a fixed $\bar{s}$, if $\sigma_q^2 \rightarrow 0$, then $P_{DF} \rightarrow 0$, as it should. Alternately, as the variance increases, the probability of a decoding failure increases.

From (V.33) with $N-K=2$, the variance of s grows linearly with N. In addition, the angle between adjacent subspaces gets smaller. Thus, as N gets large, decoding gets much more difficult, since P_{DF} depends not only upon the quantizer, but also upon the code parameters.

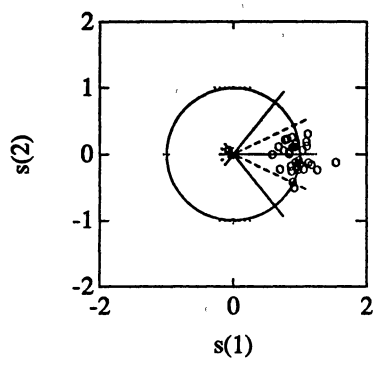
As an example, consider two BCH codes: a (7,5) and a (11,9). Figure V.3a shows a scatter plot of the two dimensional syndrome vector, $s=[s(1),s(2)]^T$ for the (7,5) code with a single error at $L=\{0\}$ of magnitude one. The codewords were quantized to four bits with a uniform quantizer optimized for a gaussian source. Figure 3b uses six bits. Figures 3c & 3d repeat the same experiment but with the (11,9) code. It's clear that there will be fewer failures for the code in 3b than in 3a since the quantization noise variance is lower. Also, comparing 3a to 3c indicates that the probability of a decoding failure for the (11,9) code will be greater than that for the (7,5) code.

P_{DF} also clearly depends upon the magnitude of $\bar{s}$. For the single error correcting case, the magnitude of $\bar{s}$ is equal to the magnitude of the error since

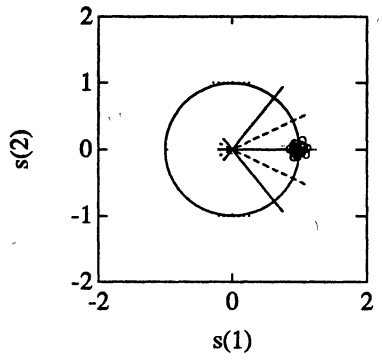
$$\| \bar{s} \| = (\bar{s}^T \bar{s})^{1/2}$$

$$= (e^T H H^T e)^{1/2}$$

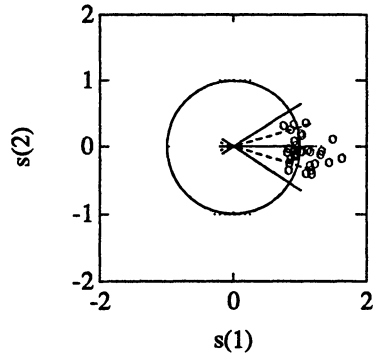
$$= \| e_{l_1} \|,$$



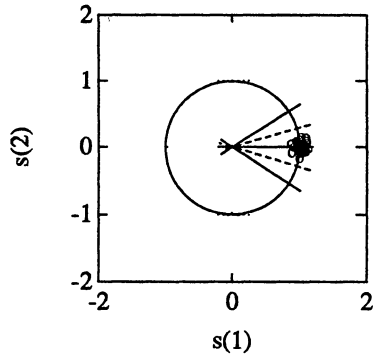
(a) (7,5) @ 4 bits



(b) (7,5) @ 6 bits



(c) (11,9) @ 4 bits



(d) (11,9) @ 6 bits

Figure V.3: Scatter Plots For BCH (7,5) and (11,9) Codes

where normalizing condition (2b) of Theorem IV.1 has been used. As e gets large, the distance from $\bar{s}$ to the next nearest subspace increases, which makes this type of error easier to decode. As e gets very small, the decoder has a more difficult time. For an arbitrarily small error, the probability of a decoding failure goes to $(N-1)/N$.

A probability of a decoding failure approaching unity is obviously not good. However, for an RN code, a decoding failure on a small error does not result in as much distortion as a failure on a large error, so this result is not so catastrophic.

If s is assumed to be normally distributed with mean and covariance given by (V.32) and (V.33) respectively, then calculating the probability of a decoding failure can be done in exactly the same way as the error rates of M-ary signal constellations are calculated, [Bla90]. However, in order to facilitate the extension of the result to the case where $t > 1$, a different approach will be taken.

Consider Figure V.4. The correct decoding region for an error with $L = \{0\}$ is given by the union of shaded region with the four square regions. By integrating the joint probability density function of the syndrome over this union, the probability of a correct decoding decision, P_c, can be found.

Approximate this union by the four square regions. Two of these squares intersect, meaning that probability of the intersection will be counted twice in the calculation of P_c. However, opposite the overlap is an area that is not counted, and due to the symmetry of the probability density function about $\bar{s}$, this area has the same probability as the region of intersection. Thus, by integrating over the region depicted by the four squares, a lower bound on the probability of a correct decoding decision can be found.

Define the *minimum angle* between syndrome error subspaces to be θ_{min}. Then for a given $\| \bar{s} \|$, the distance r is given by

$$r = \| \bar{s} \| \sin\left(\frac{\theta_{min}}{2}\right). \qquad (V.35)$$

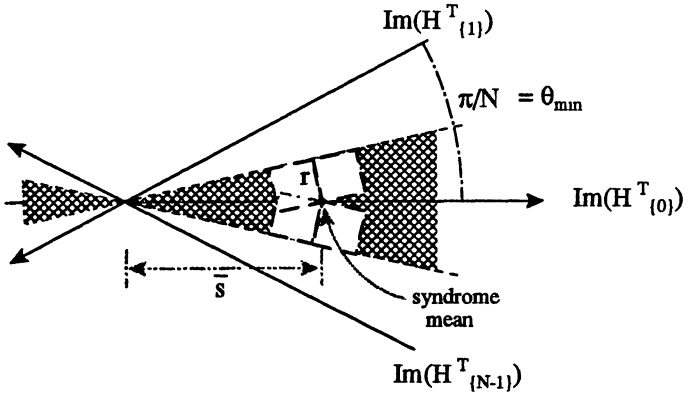


Figure V.4: Correct Decoding Region

Using (V.33) and the assumption that s is jointly normal and independent, the probability of a correct decoding decision can be written as,

$$P_C \geq \left[2 \int_0^r \frac{1}{\sqrt{2\pi\sigma_s^2}} e^{-\frac{1}{2\sigma_s^2}x^2} dx \right]^2, \qquad (V.36)$$

where

$$\sigma_s^2 = \frac{N}{N-K}\sigma_q^2.$$

Define an error function by

$$erf(r) = \frac{1}{\sqrt{2\pi}} \int_0^r e^{-\frac{1}{2}x^2} dx. \qquad (V.37)$$

The probability of a decoding failure can now be bounded as

$$P_{DF} = 1 - P_C$$

$$\leq 1 - \left[2erf\left(\frac{r}{\sigma_s} \right) \right]^2$$

$$= 1 - \left[2erf\left(\frac{\| \bar{s} \| \sin\left(\frac{\theta_{min}}{2} \right)}{\sqrt{\frac{N}{N-K}}\sigma_q} \right) \right]^2 . \qquad (V.38)$$

Since the signal to noise ratio in dB can be expressed as

$$SNR = 10\log_{10}\left(\frac{1}{\sigma_q^2} \right) , \qquad (V.39)$$

the probability of a decoding failure can be expressed as a function of the SNR,

$$P_{DF} \leq 1 - \left[2erf\left(\frac{\| \bar{s} \| \sin\left(\frac{\theta_{min}}{2} \right)}{\sqrt{\frac{N}{N-K}}10^{-\frac{SNR}{20}}} \right) \right]^2 . \qquad (V.40)$$

The reader can verify from (V.40), that for a fixed N and $\bar{s}$, as the signal to noise ratio gets large, the probability of a decoding failure goes to zero. This is what should be expected. The approximation in (V.40) is poor when $\| \bar{s} \|$ is small. Theorem V.9 summarizes this channel coding result for single error correcting codes:

THEOREM V.9: Given a single error correcting (N,N-2) BCH code with d normally distributed with $E\{d\} = 0$ and $R_d = I_K$, q distributed such that $E\{q\} = 0$ and $R_q = \sigma_q^2 I_N$; then assuming the syndrome is normally distributed about $\bar{s}$, the probability of a decoding failure can be bounded by

$$P_{DF} = 1 - \left[2erf\left(\frac{\| \bar{s} \| \sin\left(\frac{\theta_{min}}{2} \right)}{\sqrt{\frac{N}{N-K}}\sigma_q} \right) \right]^2 .$$

PROOF: Derived above.

■

Using (V.39), P_{DF} can be plotted versus SNR for different mean syndrome magnitudes. For magnitudes $\{\| \bar{s} \|\} = \{1.5, 1.25, 1.0, .75, .5, .25, .125\}$, this was done for the (7,5) and (15,13) BCH codes. The results are shown in Figures V.5 and V.6, respectively.

In an early paper, [Wol83], Wolf remarked that real number codes are tolerant of small errors (noise) on every codeword symbol. The amount of random noise that can be tolerated was the subject of future research. In Chapter I, a related question asked by Blahut, [Bla85] was presented. The question asked how small can the transmission errors become before they cannot be reliably distinguished from the random noise. Equation (V.39) provides the key to these questions for the single error correcting codes.

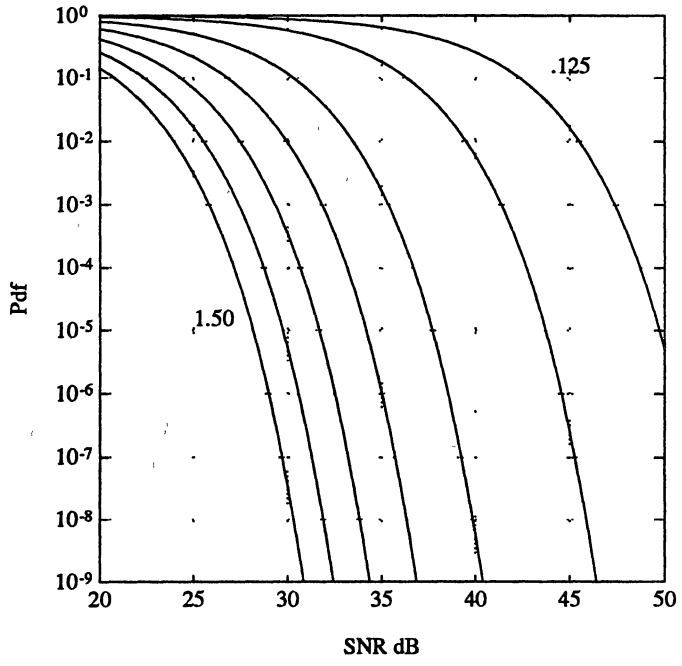


Figure V.5: P_{DF} vs. SNR for BCH (7,5) Code

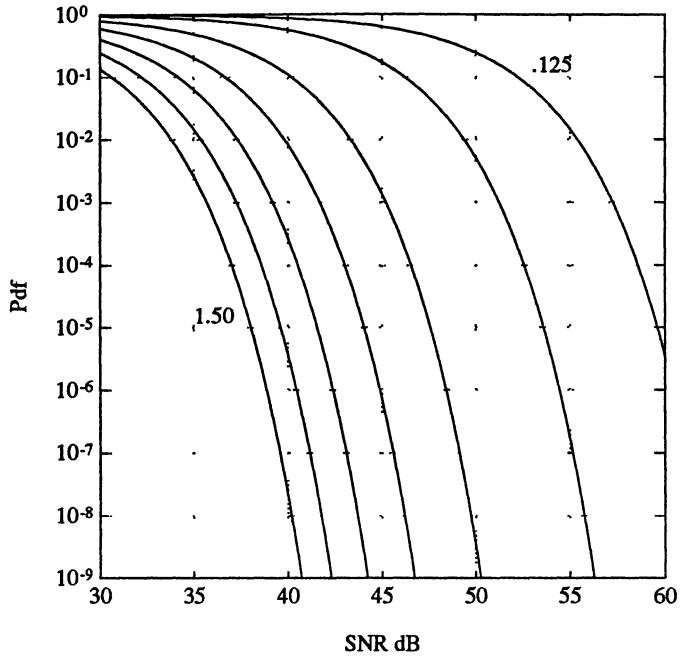


Figure V.6: P_{DF} vs. SNR for BCH (15,13) Code

For example, with a (7,5) code and a SNR of about 42 dB, transmission errors larger than .25 can be reliably corrected 999 out of 1000 times. Alternately, if one wanted to correct all errors of magnitude .25 or greater with a probability of failure less than 10^{-5}, then a SNR of about 44 dB or more is required.

Comparing Figure V.6 to Figure V.7 shows that for the (15,13) code, a higher SNR is required in order to obtain the same P_{DF}. This is what was expected, since not only does the syndrome variance increase with larger N, but also the syndrome error subspaces become more tightly packed in the syndrome space. An initial conclusion on real number codes is that they will require large signal to noise ratios combined with small block-lengths.

Figures V.6 and V.7 still do not give an average performance for these single error correcting codes. However, since the magnitude of the syndrome vector equals the error

magnitude, once a channel and quantizer are known, an average transmission error magnitude can be used to compute the average performance. For example, if a (7,5) code is used at an SNR of 40 dB and the average error magnitude is .25, then on the average, the code can be expected to correct more than 99 out of 100 single errors.

A generalization of the expressions derived in this section will be used to analyze the performance of multiple error correcting codes. However, before this is done, a general discussion on the angles between higher dimensional subspaces is needed.

Angles Between Syndrome Error Subspaces

For single error correcting codes, the minimum angle between two adjacent subspaces is both easy to calculate and visualize. The subspaces are one dimensional, so the traditional dot product relationship will give the angle.

For a general t error correcting code with a $2t$ dimensional syndrome space, the largest syndrome error subspaces will be t dimensional. One way to define angles between higher dimensional subspaces is discussed in Golub and Van Loan, [Gol83]. These angles are called the principal angles; following [Gol83] closely, they can be defined by:

DEFINITION V.10: Let A and B be subspaces in $\mathbf{R}^{N-K}$ whose dimensions satisfy

$$1 \le dim(A) = dim(B) = \mu \le t.$$

The *principal angles*, $\theta_1, \ldots, \theta_\mu \in [0, \pi/2]$, between A and B are defined recursively by

$$\cos(\theta_k) = \max_{a \in A} \max_{b \in B} a^T b = a_k^T b_k,$$

subject the constraints

$$\| a \| = \| b \| = 1,$$

$$a^T a_i = 0 \qquad i = 1, \ldots, k-1$$

$$b^T b_i = 0 \qquad i = 1, \ldots, k-1.$$

Note that $0 \le \theta_1 \le \cdots \le \theta_\mu \le \frac{\pi}{2}$. The vectors $\{a_1, \ldots, a_\mu\}$ and $\{b_1, \ldots, b_\mu\}$ are called the *principal vectors* of the subspace pair (A,B).

Definition V.10 restricts the subspaces to have the same dimension. In general, this need not be the case, but it is all that is necessary for this discussion. The principal angles and vectors can be calculated by using the SVD, [Gol83]. During this calculation it becomes clear that

$$\begin{bmatrix} a_1^T \\ \cdot \\ \cdot \\ a_\mu^T \end{bmatrix} [b_1, \ldots, b_\mu] = \begin{bmatrix} \cos(\theta_1) & \cdots & 0 \\ \cdot & & \cdot \\ \cdot & & \cdot \\ 0 & \cdots & \cos(\theta_\mu) \end{bmatrix}, \qquad (V.41)$$

where a_i and b_i are the principal vectors.

The smallest principal angle, θ_1 will be called the *minimum angle* of the subspace pair (A,B), denoted by $\theta_{min}(A,B)$. If $A \cap B = \{0\}$, then $\theta_{min}(A,B) > 0$.

The two sets of principal vectors provide orthonormal bases for their respective subspaces. Thus, a vector $a \in A$ can be written as

$$a = \alpha_1 a_1 + \cdots + \alpha_\mu a_\mu.$$

Equation (V.41) makes it clear that for any $a \in A$ and $b \in B$, the angle between a and b will be no less than θ_1 and no greater than θ_μ. An important question is if $\| a \| = \| b \| = 1$, and a, b are vectors that are uniformly distributed in A and B, respectively, what is the "average" angle between a and b?

The answer to this question cannot be easily calculated. As a rough approximation, the following average of the principal angles, θ_{avg}, seems natural:

$$\theta_{avg}(A,B) = \cos^{-1}\left(\frac{1}{\mu}\sum_{i=1}^{\mu}\cos\theta_i\right). \qquad (V.42)$$

Equation (V.42) is analogous to the source coding result of Theorem V.7, where the singular values of the pseudo-inverse were averaged. Of course, this average must be justified. Equipped with a more general definition of angles between subspaces, multiple error correcting codes can be examined.

Multiple Error Correcting Codes

The nearest subspace decoding rule chooses the μ dimensional syndrome error subspace which is closest to the syndrome vector. Again, it will be assumed that μ, the number of errors, is known and is less than t. The smaller the angles between the syndrome error subspace and adjacent subspaces, the greater the probability of a decoding failure. Arriving at the average P_{DF} for a given number of errors will require the computation of an average angle between the two closest μ dimensional syndrome subspaces as a function of μ.

Let L be a given error index set with $|L| = \mu$. For another index set J, let the principal angles be denoted by

$$\theta(Im(H_L^T), Im(H_J^T)) = \theta(L,J).$$

If $L \cap J = \varnothing$, then since $|L| = |J| \leq t$ it follows that

$$Im(H_L^T) \cap Im(H_J^T) = \{0\},$$

and $\theta_{min}(L,J) > 0$. The definition of the minimum angle as a function of the number of errors, and an average angle proceeds as follows:

DEFINITION V.11: Let L and J be error index sets with $L \cap J = \varnothing$ and $|L| = |J|$. Then define

1) $\theta_{min}(L) = \min_{J} \theta_{min}(L,J)$.

2) $\theta_{min}(\mu) = \min_{L} \theta_{min}(L) = \min_{L,J} \theta_{min}(L,J)$.

3) For the L_0 and J_0 that result in $\theta_{min}(\mu)$, let $\theta_{avg}(\mu) = \theta_{avg}(L_0,J_0)$.

The first part of Definition V.11 implies that for a fixed L, there exists a syndrome, $s \in Im(H_L^T)$, and a J such that for some $s' \in Im(H_J^T)$, the angle between s and s' is equal to $\theta_{min}(L)$.

The second part depicts the worst case. For a given μ, $\theta_{min}(\mu)$ is the minimum angle over all L and J. As N, and t get larger, it will be seen that this minimum angle can get very small, implying that for certain error patterns P_{DF} will be very large. For these directions, the syndrome error subspaces are nearly intersecting.

Using the worst case angles, the probability of a decoding failure can be bounded. By generalizing the bound on P_{DF} for the single error correcting case, an upper bound on the probability of a decoding failure as a function of the number of errors can be given as follows:

$$P_{DF}(\mu) \le 1 - \left[2erf\left(\frac{\| \bar{s} \| \sin\left(\frac{\theta_{min}(\mu)}{2}\right)}{\sqrt{\frac{N}{N-K}} 10^{-\frac{SNR}{20}}} \right) \right]^{N-K} . \qquad (V.43a)$$

Because the worst case minimum angles can become very small, the above bound is not very useful. In general, the average performance of a real number code will not be anywhere near the upper limits of this bound.

Instead of using the worst case angles, it would be desirable to have a set of "average" angles that can be used to predict the performance of multiple error correcting codes in the same way that θ_{min} is used to predict the performance of the single error correcting codes. In the third part of the definition, $\theta_{avg}(\mu)$ is defined to be the average of the principal angles using the worst case error index sets.

The true set average angles is not known. In fact, the average angles computed in (V.42) do not have any theoretical justification. One can only hope that $\theta_{avg}(\mu)$ gives a rough estimate of the true average angle. Hopefully, by using the worst case principal angles, the approximation in (V.42) will result in a conservative estimate of the true average angle, which in turn will result in a conservative estimate of the average channel coding performance.

Again, the probability of a decoding failure for single error correcting codes given in (V.40) will be generalized to the multiple error correcting case; but this time, the average angles will be used. Thus, an estimate of the average channel coding performance as a function of the number of errors is given by

$$\hat{P}_{DF}(\mu) \cong 1 - \left[2erf\left(\frac{\| \bar{s} \| \sin\left(\frac{\theta_{avg}(\mu)}{2} \right)}{\sqrt{\frac{N}{N-K}} 10^{-\frac{SNR}{20}}} \right) \right]^{N-K} . \qquad (V.43b)$$

As an example, consider the $t = 4$ BCH code of blocklength 15. It was discovered experimentally that the worst case minimum angle occurs when L and J are interleaved. For example, $L = \{0,2,4,6\}$ and $J = \{1,3,5,7\}$. Using interleaved indices, the average angles for the (15,7) code were calculated:

$$\theta_{avg}(1) = 53.3°$$

$$\theta_{avg}(2) = 42.9°$$

$$\theta_{avg}(3) = 37.4°$$

$$\theta_{avg}(4) = 33.7°$$

For comparison, the worst case angles for the (15,7) code were calculated and the angles are given by

$$\theta_{min}(1) = 53.3°$$

$$\theta_{min}(2) = 14.2°$$

$$\theta_{min}(3) = 2.40°$$

$$\theta_{min}(4) = 0.19°$$

Using (V.43a), the worst case P_{DF} was plotted versus the signal to noise ratio for the (15,7) code for the case where $\mu = 4$. Similarly, using (V.43b), the average probability of a decoding failure was plotted versus the signal to noise ratio for the (15,7) code. Again, four errors were assumed. Figures V.7 and V.8, respectively, show the results.

Unlike the $t = 1$ case, when $\mu > 1$, the magnitude of the error vector is not equal to the magnitude of the syndrome mean, since

$$\begin{aligned} \| \bar{s} \| &= (\bar{s}^T \bar{s})^{1/2} \\ &= (e^T H H^T e)^{1/2} \\ &\neq \| e \| . \end{aligned}$$

However, in general, it can be assumed that larger magnitude errors will produce larger magnitude syndrome means.

Some simulation results for the (15,7) code are included in Chapter VII. These results examine the probability of a decoding failure for a fixed error position that corresponds to the worst case.

Also in Chapter VII, a second simulation uses random error locations with random magnitudes. This simulation gives an indication of the average channel coding performance of the (15,7) code and can be compared against the estimated average performance.

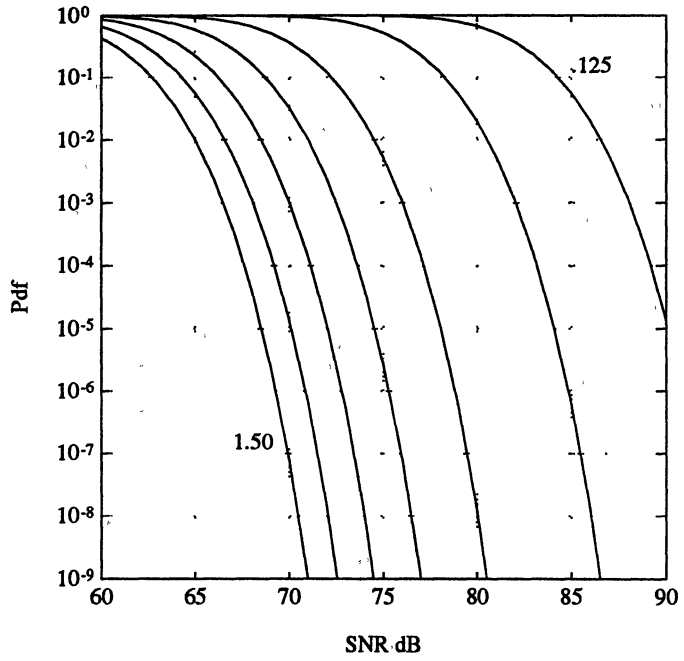


Figure V.7: Worst Case P_{DF} For (15,7) Code With t=4 Errors

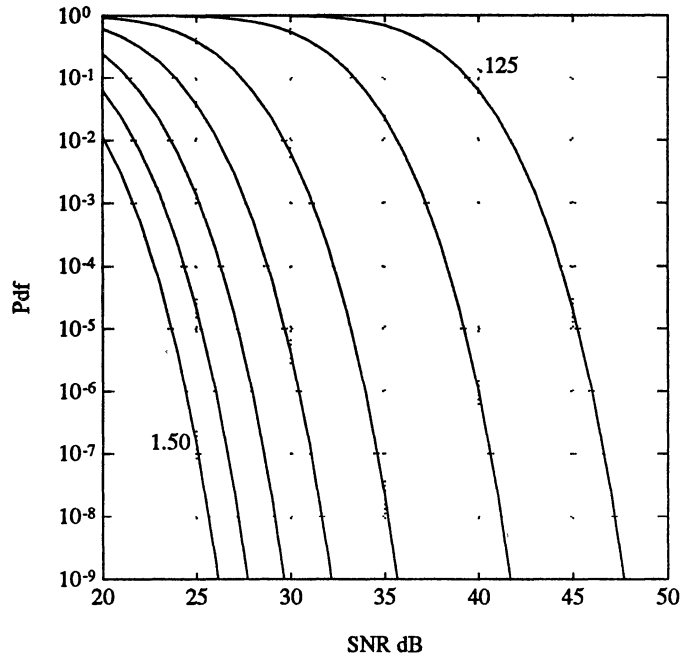


Figure V.8: Average P_{DF} For (15,7) Code With t=4 Errors

Joint Source-Channel Coding

Recall that the total mean squared error for a finite field based system could be written as

$$MSE_{FF} = M1_{FF}(1-P_E)+M2_{FF}P_E \qquad (V.44)$$

where $M1_{FF} = \sigma_q^2$ and $P_E = P_{UNC}$. For the real number system,

$$MSE_{RN} = M1_{RN}(1-P_E)+M2_{RN}P_E. \qquad (V.45)$$

In this latter case, $M1_{RN}$ is a function of the number and location of the errors, the code parameters, and the quantizer. Also, $P_E = P_{UNC} + P_{DF}$, where P_{UNC} is the probability of an uncorrectable error and P_{DF} is the probability of a decoding failure.

It was seen that similar to how $M1_{RN}$ characterizes the source coding performance of an RN code, the channel coding performance of a real number code is characterized by P_{DF}. P_{DF} is also a function of the number and location of the errors, the code parameters, and the quantizer.

In order to get an average source coding performance, $M1_{RN}$ as a function of L (denoted by $M1_{RN}(L)$) can be averaged over all possible index sets of a given order to get $M1_{RN}(|\,L\,|)$. (Assume that each L is equally likely.) This was done for two $t = 4$ BCH codes. The first is the (15,7) code while the second is the (19,11) code. The results are plotted in Figure V.9.

The plot shows $M1_{RN}/M1_{FF}$ in dB as a function of $|\,L\,|$. The dashed 0 dB line represents the performance of an equivalent finite field code. The regions where the real number values lie below this line indicate that the mean squared error of the real number code is superior to the finite field code.

In order to compare the total performance of the finite field system to the real number system, (V.44) and (V.45) must be used to compute MSE_{FF} and MSE_{RN}. This is a very difficult problem, since $M2_{FF}$ and $M2_{RN}$ are unknown. In addition, P_{DF} is a function of not only the number and locations of the errors, the code parameters, and the quantizer, but it is also a function of the magnitude of the syndrome vector.

To avoid these problems, it will be assumed that the systems have been designed so that P_E is sufficiently small so that (V.44) and (V.45) reduce to

$$MSE_{FF} \cong \sigma_q^2 \qquad (V.46)$$

$$MSE_{RN} \cong M1_{RN}(|\,L\,|). \qquad (V.47)$$

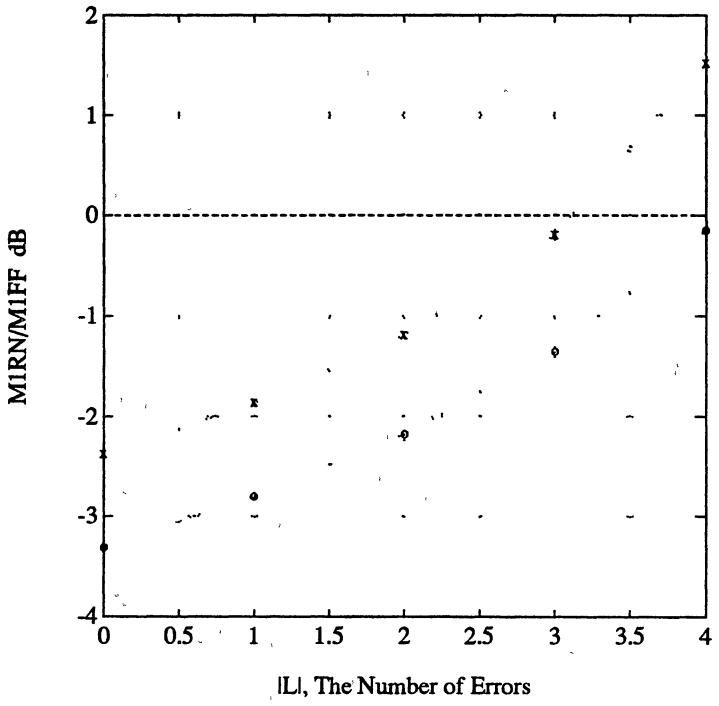


Figure V.9: Average Source Coding Performance For Two BCH Codes:

o - (15,7) code, x - (19,11) code

dashed - Performance for FF Code

Next, a channel model must be specified. Since Reed Solomon and BCH codes correct at the symbol level, choose a memoryless channel where the probability of a symbol error is given by P_S. Since N is the blocklength, anywhere from 0 to N symbol errors can occur. The probability of μ errors is given by

$$prob_{|L|}(\mu \text{ symb. errors}) = \binom{N}{\mu}(1-P_S)^{N-\mu}P_S^{\mu}. \qquad (V.48)$$

For (V.46) and (V.47) to hold, P_S must be chosen so that P_E is small. If P_S is restricted such that $P_S \leq .015$, then for the (15,7) and the (19,11) codes,

$$P_{UNC} = \sum_{\mu=t+1}^{N} P_{|L|}(\mu) \leq 10^{-5}. \qquad (V.49)$$

With these assumptions, MSE_{RN} relative to MSE_{FF} in dB for the (15,7) and the (19,11) codes are plotted versus P_S in Figure V.10.

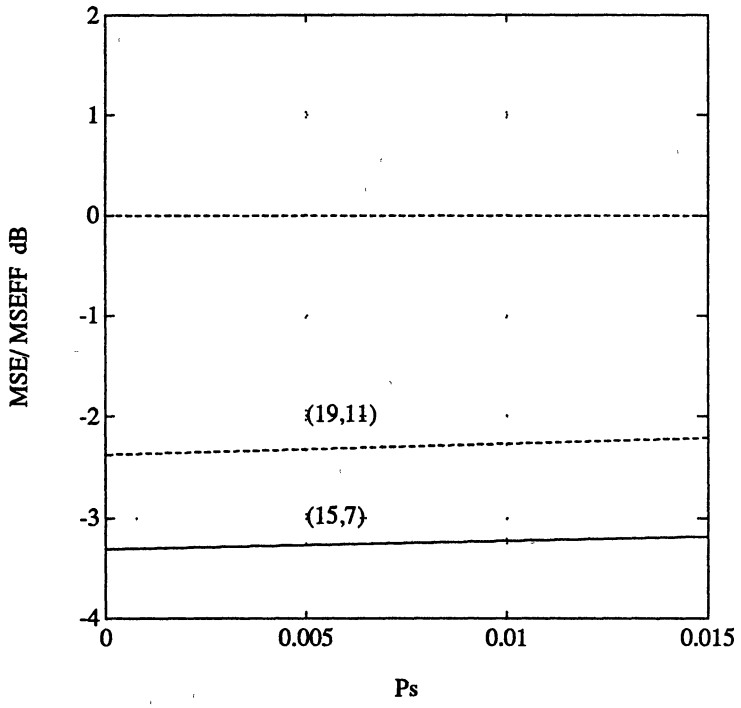


Figure V.10: MSE_{RN} versus P_S for (15,7) and (19,11) Codes.

Figure V.10 suggests that real number codes could have some advantages over the finite field codes. However, two big questions remain. First, is under what conditions is P_{DF} negligible so that $P_E \approx P_{UNC}$?, (if ever). Second, given these conditions, are there any realistic applications that could benefit from an RN code.

Unfortunately, P_{DF} is never really negligible. There always exist error patterns that will not be reliably corrected. In addition, these error patterns are large enough in magnitude such that the resulting distortion due to incorrectly decoding the information word is significant. These errors will increase MSE_{RN} relative to MSE_{FF}, and so the results in Figure V.10 are very optimistic.

The results in this chapter were derived by assuming that the number of error is known. In Chapter VI, a general decoding strategy for RN codes is presented that does not assume any information regarding the number of errors. This strategy is based upon trying to minimize MSE_{RN}. Then in Chapter VII, several simulations are presented, including source coding, channel coding and joint source-channel coding simulations using the decoding method of Chapter VI. This final simulation perhaps provides the best indicator for the complete joint source-channel performance of RN codes.

CHAPTER VI

DECODING RN BCH AND RS CODES IN ADDITIVE NOISE

In Chapter V, the discussion on the nearest subspace decoding rule assumed that the number of errors was known. However, this is not the case. Recall, that for the infinite precision case, the rank of the syndrome matrix, (Theorem IV.4), gave the number of errors. For the noisy case, in general, the syndrome matrix will be of full rank due to the presence of quantization noise. Rather than concluding that there are always t errors, decoding methods must either estimate the number of errors or proceed without this information.

Methods for estimating the number of errors will be exactly the same as the methods for estimating the number of complex sinusoids in the signal processing literature. However, instead of delving into the sinusoid problem, an overview of some possible approaches to the noisy decoding problem is presented. Most of these approaches draw upon the estimation of sinusoids or simple the sinusoid estimation techniques, and have been explained in more detail by previous papers on real number error correction codes. In a sense, the next section is a continuation of the "Previous Work" section in Chapter I; since at that point, a detailed discussion of these decoding methods was not appropriate.

Decoding Methods

The first decoding technique is the previously mentioned "voting argument", [Wol83b]. It does not assume that the number of errors is known, nor does it rely on an

estimate. Wolf used this argument to assert that real number codes could possibly correct up to twice as many errors as finite field codes. Specifically, it was proposed that real number codes could correct up to N-K-1 errors. The argument goes as follows.

The encoding equation,

$$\begin{bmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_N \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1K} \\ g_{21} & g_{22} & \cdots & g_{2K} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ g_{N1} & g_{N2} & \cdots & g_{NK} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ d_K \end{bmatrix}, \qquad (VI.1)$$

represents a system of N equations with K unknowns. Now consider that there exist μ errors in the received word, with $\mu \leq N-K-1$. By Theorem III.7, it is known that any subset of K equations will give a unique estimate for the information word.

Suppose that each of the "N choose K" possible estimates are calculated and are represented by a point in a K dimensional space. At the least, there are

$$N-(N-K-1)=K+1$$

equations that are not in error. This results in $K+1$ estimates of the information word, which given infinite precision, are equal. If the codeword has been quantized, then there will be a cluster of $K+1$ points around the true information word.

Given that the transmission errors are random and are chosen from a continuum of possible values, it is unlikely that more than $K+1$ of the remaining estimates will be equal, or clustered. Thus, the decoding method would be to pick the point in K-space which is the centroid of such a cluster. Of course, there exist error patterns with weight greater than t, where the decoding will fail. For $\mu \leq t$, it was seen that the true data estimate can always be found regardless of the error pattern.

This argument is not limited only to codes over the real or complex fields. Finite field codes could be decoded this way. The trouble for finite field codes, especially fields with a small number of elements, is that the number of possible error patterns is limited, thus making the decoding method less reliable for $\mu > t$ errors.

For example, look at a (5,1) repeat code. There are 5 estimates of the data symbol, which correspond to the five received symbols. This code can always correct up to two errors, since at least 3 of the data estimates will be equal. If there are $N-K-1=3$ errors, then unless two of the errors are equal, there is a cluster of 2 points and the data symbol can be recovered. Four errors can never be corrected.

Clearly, if the field is small, this argument breaks down. Three errors can never be corrected with a (5,1) binary repeat code, since the error are always equal. However, the larger the field, the more unlikely it is that two errors are equal.

The problem with such a decoding method is the large number of calculations. It requires the solution of

$$\binom{N}{K}=\frac{N!\,K!}{(N-K)!},$$

$K \times K$ systems of equations. In addition, if the codeword was quantized, then determining when a point belongs in a particular cluster instead of another becomes a problem. Again, the decoding accuracy will depend upon the quantization noise level. Instead of using this approach, nearest subspace decoding methods and techniques which rely on the structure of the parity check matrix will be pursued.

Because of the Vandermonde structure of the parity check matrix, it was possible to locate the errors using Prony's method. This technique worked perfectly in the noiseless case; however, in the noisy case, the performance Prony's method can be rather poor. In the estimation literature, several modifications to the basic Prony method have been developed which improve its performance in the presence of noise. Three of these modi-

fications are discussed for use with RN BCH and RS codes.

Recall, that Prony's method required knowledge of the number of errors. This information was given by the rank of the syndrome matrix. For the noisy case, the *approximate rank* of the syndrome matrix can be estimated with the singular value decomposition. Technically, the noisy syndrome matrix will be full rank, that is, none of the singular values of this matrix will be zero. However, for μ errors, if the noise is sufficiently small compared to the magnitudes of the errors, then the first μ singular values will be noticeably larger than the others. Thus, the approximate rank of the syndrome matrix is μ.

Using Theorem B.2, a *reduced rank approximation* to the syndrome matrix can be found. Then the first modification is to use the reduced rank syndrome matrix in place of the original syndrome matrix in Prony's method. This procedure is standard in the estimation literature; it has been proven to reduce the effects of the noise.

A second noise reducing addition to Prony's method can be performed when $\mu < t$. In this case, by utilizing all the syndromes, Equation (IV.10) becomes

$$-\begin{bmatrix} S_{\mu} \\ S_{\mu+1} \\ \cdot \\ \cdot \\ \cdot \\ S_{2t-1} \end{bmatrix} = \begin{bmatrix} S_0 & S_1 & \cdots & S_{\mu-1} \\ S_1 & S_2 & \cdots & S_{\mu} \\ \cdot & & \cdot & \\ \cdot & & \cdot & \\ \cdot & & \cdot & \\ S_{2t-\mu-1} & S_{\mu} & \cdots & S_{2t-2} \end{bmatrix} \begin{bmatrix} \alpha_{\mu} \\ \alpha_{\mu-1} \\ \cdot \\ \cdot \\ \alpha_1 \end{bmatrix}. \qquad (VI.2)$$

Equation (VI.2) is a set of overdetermined equations. When solving (VI.2), a least squares solution can be employed. The least squares solution tends to average out the effects of the quantization noise. In addition, the least squares solution can be used in conjunction with the reduced rank approximation to further reduce the noise effects. These methods are essentially the methods employed by Kumaresan, [Kum85].

The final modification found in the sinusoid estimation problem is to overestimate the order of the predictor. In the error correction problem, this corresponds to overestimating the order of the error locator polynomial. In the sinusoid literature, it was found that Prony's method performs poorly for closely spaced sinusoids. By using a larger predictor order, it was found that better resolution between the sinusoid frequencies can be obtained. The trade-off is that "spurious" frequencies exist which correspond to the extra roots of the prediction polynomial. As a consequence, one must decide which frequencies are spurious artifacts of the noise and which are due to sinusoidal components in the signal.

All three of these modifications have been used with great success in the sinusoid estimation problem. In the error location problem, these modifications are less useful since fewer data points are available. As an example, in a typical sinusoid estimation problem with two sinusoids in noise, 25 or 35 data points might be available. For this case, a prediction order of ten or higher might be utilized, [Tuf82]. For the error correction problem, there are not enough syndrome values available to overestimate the maximum number of errors. With $\mu = t$, (VI.2) is a square system of equations; there cannot be any over estimation.

In the error correction problem, $2t$ syndromes are available to estimate up to t errors. If there are t errors, then none of the above enhancements to Prony's method can be applied.

A final note concerning application of Prony's method to the noisy error correction problem is the following. The final result of Prony's method is a set of complex roots. Ideally, the magnitudes of these roots are one, and the angles are from a discrete set which correspond to the error locations. With noise however, Prony's method is not restricted to discrete angles on the unit circle. For this reason, the angles must be rounded to discrete values. As a rule of thumb, since Prony's method performs poorly

for closely spaced sinusoids, in the case where two angles are rounded to the same discrete value, it is advantageous to split the two so that consecutive locations are chosen as error location estimates.

Decoding strategies based upon Prony's method are suboptimal. It was stated in Chapter V, that the nearest subspace decoding rule corresponds to the optimal rule. However, a problem in using the nearest subspace decoding (NSD) rule is that it too relies upon the knowledge of how many errors are present..

One approach would be to calculate the SVD of the syndrome matrix to get an estimate of the number of errors. In general, this can be difficult since it implies that a threshold for the singular values must be found. The singular values nonlinearly depend upon the error magnitudes and the noise level. Finding a threshold which is robust to the varying ranges of error magnitudes, but yet conservatively estimates the number of errors is difficult. Therefore, instead of estimating the rank, two other approaches based upon the NSD rule will be presented.

The first approach is to find the nearest syndrome subspace for every possible error number. That is, decode the syndrome by first assuming one error, then two errors, etc., all the way up to t errors. This will be called the *brute force* NSD, and it is essentially the method presented in [Sch87]. This is obviously a computationally intensive procedure. It first starts by assuming one error, and the distance from the noisy syndrome to all one dimensional syndrome error subspaces is then computed. The location and value corresponding to the minimum of all these distances is then recorded. Next, two errors are assumed and a search over all possible two dimensional syndrome error subspaces is performed. This procedure continues until a full search over all possible t dimensional subspaces has been completed. This final step is an optimization problem in a t dimensional discrete space.

As the assumed number of errors increases, the distance from the syndrome to the nearest syndrome error subspace will always decrease. From the results in Chapter V, if there are fewer errors, a more accurate data word estimate results. Thus, the goal in this decoding procedure is to keep the estimated number of errors to a minimum while ensuring that all true error locations have been found. One way to do this is to examine the recorded minimum distance as a function of the assumed number of errors. The minimum distance should be relatively large and decreasing steadily until the assumed number of errors equals the actual number of errors. After this point, the minimum distance will tend to level out.

For example, consider a the (15,7) BCH with error index vector, $L = \{0, 1, 3\}$ and error magnitudes {.5, 1.0, -.25} respectively. For a randomly generated Gaussian quantization noise vector, the brute force NSD algorithm was performed. The SNR was 30 dB. Figure VI.1 shows the minimum syndrome subspace distance as a function of the assumed number of errors.

The dashed line represents the expected value of the norm of the syndrome vector due to quantization alone (assuming that the quantization error is white). This value is given by

$$\begin{aligned} E\{\| H^T q \|\} &= E\{\sqrt{q^T H H^T q}\} \\ &= \sqrt{N\sigma_q^2} \\ &= \sqrt{N}\,\sigma_q \\ &= 0.123. \end{aligned} \qquad (VI.3)$$

For the above example, the nearest syndrome error subspaces where given by {1}, {0, 1}, {0, 1, 3}, {0, 1, 3, 7}, by assuming $\mu = 1, 2, 3, 4$ errors, respectively. Assuming no errors, the distance is merely the norm of the syndrome. As the assumed number of errors increases, the distance decreases. Once three errors are assumed, the nearest subspace corresponds to the error index vector, so in this case, the errors are correctly

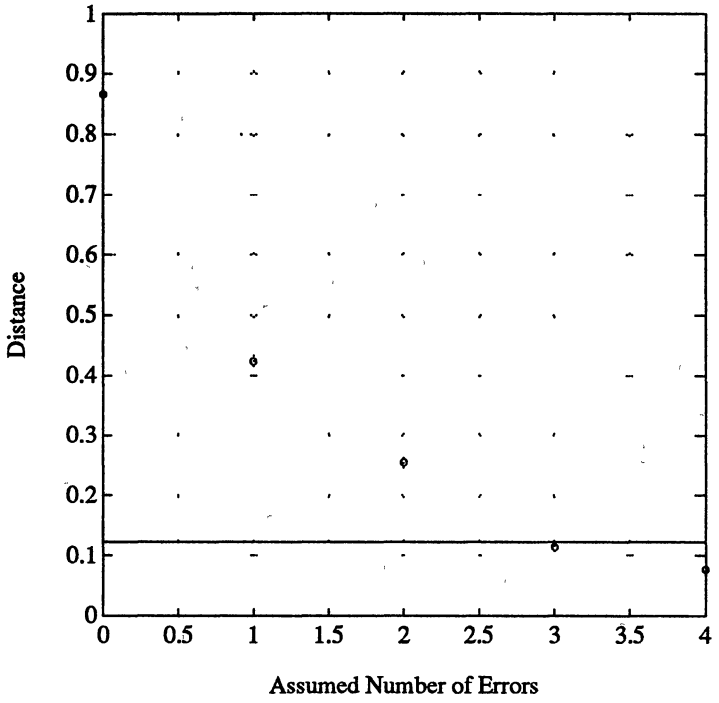


Figure VI.1: Minimum Syndrome Subspace Distance vs. Assumed Number of Errors

located. The results in Figure VI.1 give a fair indication that the number of errors is 3 since at that point, the distance dips below the expected value of the syndrome due to noise alone.

In general, a threshold for the distance will be chosen which will indicate the number of errors. From (VI.3), this threshold must clearly be a function of the quantization noise level and the blocklength.

As Scharf and his colleagues point out [Sch87], this method is computationally intensive since a full search over all syndrome error subspaces for every possible number

of errors is performed. For large codes, the technique is impractical. As a possible remedy, they suggest that Prony's method might be used as a starting point for a limited search decoding method.

Using this idea, the author has developed an approximate NSD method that attempts to combine the subspace search method with Prony's method. This method is explained in the next section. In Chapter VII, some simulation results obtained with this method are presented.

Approximate NSD Method

Rather than computing a search for every possible number of errors, the approximate NSD (ANSD) method starts by assuming that there are t errors. After which, the ANSD method essentially consists of two steps:

(1) Locate an error index set of order t which is believed to correspond to the nearest syndrome subspace. This is accomplished by using a limited search about a starting point.

(2) Attempt to reduce the order of the index set by using a cost comparison procedure.

Prony's method is used to obtain the starting point. By assuming that there are t errors, the largest possible order for the error locator polynomial has been obtained. Hopefully, this will enhance the performance of Prony's method.

The result of using Prony's method is an error index set which is a point,

$$L^{(0)} = \{l_1, \ldots, l_t\} \in [0, \ldots, N-1] \times [0, \ldots, N-1] \times \cdots \times [0, \ldots, N-1], \qquad (t \text{ times})$$

with the only constraint being that all $\{l_i\}$ are distinct. (Relax the constraint that the error index set is ordered.) This point represents a syndrome subspace, but not necessarily the nearest syndrome subspace. Only in the case where $t = 1$, can it be shown that the NSD and Prony's method give the same result.

For $t > 1$, the approximate NSD algorithm assumes that $L^{(0)}$ is "close" to the nearest syndrome subspace. Of course, "close" must be defined. Consider $L = \{l_1, \ldots, l_\mu\}$ and $L^{(0)} = \{l_1^{(0)}, \ldots, l_\mu^{(0)}\}$ as points in $(\mathbf{Z}_{N-1})^t$. The distance between l_1 and $l_1^{(0)}$, denoted by $| l_1 - l_1^{(0)} |_N$, is *circular modulo N*. That is, define

$$| l_1 - l_1^{(0)} |_N = \min | l_1 - l_1^{(0)} + kN |, \quad k = -1, 0, 1. \tag{VI.4}$$

The term "circular" comes from the analogy of N roots of unity on the unit circle. One could define a distance between any two roots to be the integer number of angular steps required to go from one root to the other. For example, if $N = 7$, then

$$| 0 - 6 |_7 = \min(13, 6, 1)$$
$$= 1.$$

Using this distance between the elements of an index set, one can define a discrete distance measure between L and $L^{(0)}$ as

$$d_{dis}(L, L^{(0)}) = \max(| l_1 - l_1^{(0)} |_N, \ldots, | l_t - l_t^{(0)} |_N). \tag{VI.5}$$

Two index sets will be considered to be "close", when the discrete distance measure in (VI.5) is small. The index set computed with Prony's method will be used as the starting point for a local search.

Associated with each index set is the real number

$$F(L) = \| (I - P_L)s \|, \tag{VI.6}$$

where P_L is the orthogonal projection onto the syndrome subspace corresponding to index set L. $F(L)$ is the distance from the syndrome to the syndrome subspace that corresponds to L. From Chapter V, finding the nearest subspace is the same as minimizing $F(L)$ with respect to L.

Define a *discrete neighborhood of radius one about* $L^{(0)}$ to be the set of all index sets, L, such that $d_{dis}(L, L^{(0)}) \leq 1$. To start the local search, compute $F(L)$ for each L in the

discrete neighborhood of radius one about $L^{(0)}$. For some $L^{(1)}$ in this neighborhood, $F(L)$ is minimum. If $L^{(1)} = L^{(0)}$, then the search is over and the initial Prony's estimate is assumed to be the NSD estimate.

If $L^{(1)} \neq L^{(0)}$, then the search can be continued using $L^{(1)}$ as the center for a new discrete neighborhood. Note that since

$$d_{ds}(L^{(0)}, L^{(1)}) \leq 1,$$

the old and new discrete neighborhoods will overlap. Thus, the computations to compute $F(L)$ for the new discrete neighborhood can be reduced.

The maximum number of iterations will depend upon the amount of search time that is available. The performance of the method depends heavily upon the initial accuracy of $L^{(0)}$. If this starting point is far away from the true minimum, then either the search will require many iterations, or a minimum local to the discrete neighborhood will be found and a decoding failure might result.

Since the number of errors is not always equal to t, it will be necessary to redefine what constitutes a decoding failure. Suppose L is the true error index set with $|L| = \mu$. Let J_0 be the result obtained from the ANSD search, so $|J_0| = t$. If $L \subset J_0$, then the syndrome has been correctly decoded. Otherwise, a decoding failure has occurred.

For example, with the (15,7) example in the previous section, the error locations were given by $L = \{0, 1, 3\}$. The minimum t-dimensional subspace was given by $J = \{0, 1, 3, 7\}$. Since $L \subset J$, the decoder did not fail. Using the ANSD with a limited search, it is possible that $J_0 = \{0, 1, 3, 12\}$ could result. This result is still a distance of 5 away from J, but yet the error index set has been correctly decoded since $L \subset J_0$.

The conclusion of the local search marks the end of the first step in the ANSD process. The second step is to attempt to reduce the order of the resulting order t error index set.

Once the error locations have been found, the information word can be estimated. However, the error index set obtained from the ANSD search is of order t. Since $M1_{RN}$ increases as a function of the number of errors, it is beneficial to reduce the order of the index set by keeping only those indices that correspond to transmission errors and not those due to quantization noise.

Intuitively , it is easy to see that $M1_{RN}$ increases, since the fewer equations deleted in the estimation of the information word, the better the least squares estimate. More formally, by using the results of Theorem V.7 combined with Theorem B.4 it can be shown that $M1_{RN}$ increases. Therefore the second step in the ANSD procedure is to systematically reduce the order of the error index set. The reduction procedure goes as follows.

First, given the order t error index set $J = \{j_1, \ldots, j_t\}$, which was obtained from the search procedure, compute an estimate of the error amplitudes. This is done in the same fashion as described by (V.17). The result is a set of error values,

$$\{e_{j_1}, \ldots, e_{j_t}\}.$$

Denote the one with smallest magnitude by $e_{min}(t)$. The question that needs to be answered is whether $e_{min}(t)$ is truly a transmission error or is it an artifact of the quantization noise? In order to determine this, $| e_{min}(t) |$ must be compared against a threshold value, denoted by $\xi(t)$.

If $| e_{min}(t) | > \xi(t)$, then the error is genuine and the error index set is kept intact. If $| e_{min}(t) | \leq \xi(t)$, then it is deduced that $e_{min}(t)$ is an artifact of the quantization noise, and the index corresponding to the error amplitude is deleted from the error index set. Now the order of the index set is $t - 1$, and the procedure is repeated by computing a new set of error amplitudes.

In general, t thresholds must be calculated. The general reduction rule is as follows:

ANSD Reduction Rule: For a given error index set, J, of order μ, compute the set of error amplitudes and locate the one with smallest magnitude. Denote this magnitude by $|e_{min}(\mu)|$. Let j_{min} be the corresponding index. If

$$|e_{min}(\mu)| \leq \xi(\mu) \quad \text{then}$$

$$J = J - \{j_{min}\},$$

else J is unchanged and the procedure stops.

After the error index set has been completely reduced, the final information word can be estimated and the ANSD procedure is complete. The only task that remains is to derive a set of thresholds.

A set of thresholds can be determined by comparing the costs of reducing or not reducing the error index set. These costs will be measured in terms of the mean squared error. Consider the first reduction decision pertaining to $e_{min}(t)$. Four cases exist:

1a) (Type I Error) The index j_{min} does not correspond to an error and the decision is to keep $j_{min} \in J$.

1b) (Correct) The index j_{min} does not correspond to an error and the decision is to reduce J, i.e. $J = J - \{j_{min}\}$.

2a) (Correct) The index j_{min} corresponds to an error and the decision is to keep $j_{min} \in J$.

2b) (Type II Error) The index j_{min} corresponds to an error and the decision is to reduce J.

Calculate the average cost in making a type I error. In case 1a, the information word will be estimated using an error index set of order t. In case 1b, an order $t-1$ index set is used. Therefore the cost is given by

$$C_I = E\left\{\frac{1}{K}(d-\hat{d})^T(d-\hat{d}) \;:\; |J|=t\right\}$$
$$-E\left\{\frac{1}{K}(d-\hat{d})^T(d-\hat{d}) \;:\; |J|=t-1\right\}. \qquad (VI.7)$$

In (VI.7), the mean squared error terms will depend upon the singular values, as shown in Theorem V.7. When calculating the cost, two approaches can be taken. First, the singular values corresponding to a particular J can be used to find the MSE terms; or second, an average MSE as a function of $|J|$ can be used, similar to what is depicted in Figure V.9.

The first approach will result in a threshold for each possible J, while the second approach results in one set of t thresholds. In this discussion, the second approach is taken.

For convenience, explicitly write $M1_{RN}$ as a function of J,

$$Ml_{RN}(J) = \sigma_q^2 \frac{1}{K}\sum_{i=1}^{K}\frac{1}{\sigma_i^2(J)}. \qquad (VI.8)$$

Now by averaging over all possible J with a fixed order, write

$$Ml_{RN}(|J|) = \sigma_q^2 E\left\{\frac{1}{K}\sum_{i=1}^{K}\frac{1}{\sigma_i^2(J)}\right\}$$
$$= \frac{\sigma_q^2}{\sigma_{avg}^2(|J|)}, \qquad (VI.9)$$

where

$$\frac{1}{\sigma_{avg}^2(|J|)} = E\left\{\frac{1}{K}\sum_{i=1}^{K}\frac{1}{\sigma_i^2(J)}\right\}. \qquad (VI.10)$$

Using (VI.9), the cost of a type I error can now be written as

$$C_I = Ml_{RN}(t) - Ml_{RN}(t-1). \qquad (VI.10)$$

Since $Ml_{RN}(|J|)$ is increasing, C_I is positive.

Now calculate the cost of making a type II error. The cost of a type two error depends upon $|e_{\min}|$ since the corresponding error location is incorrectly being deleted from the error location set. It is the difference between the mean squared error of cases 2b and 2a, written as

$$C_{II} = E\left\{\frac{1}{K}(d-\hat{d})^T(d-\hat{d}) \;:\; |J| = t-1, |e_{\min}|\right\}$$
$$-E\left\{\frac{1}{K}(d-\hat{d})^T(d-\hat{d}) \;:\; |J| = t\right\}. \qquad (VI.11)$$

Isolate the first expectation. That is,

$$E\{MSE : |J| = t-1, e_{\min}\} = E\left\{\frac{1}{K}(q_{J^c} + S_{J_{\min}} e_{\min})^T \left(G_{J^c}^+\right)^T \left(G_{J^c}^+\right)(q_{J^c} + S_{J_{\min}} e_{\min})\right\}$$
$$= E\left\{\frac{1}{K} q_{J^c}^T \left(G_{J^c}^+\right)^T \left(G_{J^c}^+\right) q_{J^c}\right\}$$
$$+\frac{1}{K} e_{\min} S_{j_{\min}}^T E\left\{\left(G_{J^c}^+\right)^T \left(G_{J^c}^+\right)\right\} S_{J_{\min}} e_{\min}.$$

Using the Theorem V.7,

$$E\{MSE : |J| = t-1, e_{\min}\} = Ml_{RN}(t-1)$$
$$+\frac{1}{K} e_{\min} S_{J_{\min}}^T E\left\{\left(G_{J^c}^+\right)^T \left(G_{J^c}^+\right)\right\} S_{J_{\min}} e_{\min}$$
$$\cong Ml_{RN}(t-1) + \frac{1}{N}\frac{(e_{\min})^2}{\sigma_{avg}^2(t-1)}. \qquad (VI.12)$$

In deriving (VI.12), the quantization noise and the transmission errors are assumed to be independent. If the SVD of the deleted generator matrix is written as

$$G_{J^c} = U\Sigma V^T,$$

and $x = S_{J_{min}} e_{min}$, then the final approximation in (VI.12) is from approximating

$$x^T E\left\{\left(G_{J^c}^+\right)^T\left(G_{J^c}^+\right)\right\}x = x^T E\{U(\Sigma^+)^T(\Sigma^+)U^T\}x$$

$$\cong x^T U' \begin{bmatrix} \frac{1}{\sigma_{avg}^2(t-1)} I_K & 0 \\ 0 & 0 \end{bmatrix} (U')^T x$$

$$\cong \frac{K}{N}\frac{(e_{min})^2}{\sigma_{avg}^2(t-1)}.$$

Thus the cost of making a type II error can be approximated as

$$C_{II} \cong Ml_{RN}(t-1) - Ml_{RN}(t) + \frac{1}{N}\frac{(e_{min})^2}{\sigma_{avg}^2(t-1)}$$

$$= -C_I + \frac{1}{N}\frac{(e_{min})^2}{\sigma_{avg}^2(t-1)}. \qquad (VI.13)$$

If $C_I > C_{II}$ then making a type II error is not as costly as making a type I error, so the error index set should be reduced. Alternately, if $C_{II} > C_I$, then the error index set should remain intact.

To compare C_I vs. C_{II}, evaluate $C_{II} - C_I$. Using (VI.9),

$$C_{II} - C_I = \frac{1}{N}\frac{(e_{min})^2}{\sigma_{avg}^2(t-1)} - 2C_I$$

$$= \frac{1}{N}\frac{(e_{min})^2}{\sigma_{avg}^2(t-1)} - 2(Ml_{RN}(t) - Ml_{RN}(t-1))$$

$$= \frac{1}{N}\frac{(e_{min})^2}{\sigma_{avg}^2(t-1)} - 2\sigma_q^2\left(\frac{1}{\sigma_{avg}^2(t)} - \frac{1}{\sigma_{avg}^2(t-1)}\right) \qquad (VI.14)$$

So $C_{II} - C_I > 0$ whenever

$$(e_{min})^2 > 2N\sigma_q^2\left(\frac{\sigma_{avg}^2(t-1)}{\sigma_{avg}^2(t)} - 1\right). \qquad (VI.15)$$

By letting

$$\xi^2(t) = 2N\sigma_q^2\left(\frac{\sigma_{avg}^2(t-1)}{\sigma_{avg}^2(t)} - 1\right)$$

$$= 2N\sigma_q^2\left(\frac{M1_{RN}(t)}{M1_{RN}(t-1)} - 1\right), \qquad (VI.16)$$

a threshold has been derived such that if $|\, e_{\min} \,| > \xi(t)$, then on the average, $C_{II} > C_I$, which implies that the error index set should remain intact. From (VI.16), since $M1_{RN}$ is increasing, the threshold is never complex. This should be expected.

By following the same procedure, a complete set of thresholds can be derived. If $e_{\min}$ is the smallest magnitude error amplitude with $|\, J \,| = \mu$, then let

$$\xi^2(\mu) = 2N\sigma_q^2\left(\frac{M1_{RN}(\mu)}{M1_{RN}(\mu-1)} - 1\right), \qquad (VI.17)$$

for $\mu = t, \ldots, 1$.

From (VI.17), it is seen that the average thresholds are a function of the blocklength and the quantization noise level. Again, this would be expected. Using the information obtained from Figure V.9, the thresholds can be calculated for the (15,7) BCH code as follows:

$$\xi(1) = 1.90\sigma_q$$

$$\xi(2) = 2.15\sigma_q$$

$$\xi(3) = 2.50\sigma_q$$

$$\xi(4) = 3.10\sigma_q.$$

Similarly, the thresholds for the (19,11) code,

$$\xi(1) = 2.18\sigma_q$$

$$\xi(2) = 2.53\sigma_q$$

$$\xi(3) = 3.14\sigma_q$$

$$\xi(4) = 4.28\sigma_q.$$

The steps of the ANSD algorithm can be summarized as follows:

1) Compute the initial order t index set by using Prony's method and rounding the root locations to the nearest roots of unity. The locations should be altered if there are repeated roots. Denote this set by $L^{(0)}$.
2) For each index set L within a discrete neighborhood of radius one about $L^{(0)}$, compute F(L) using (VI.6).
3) Select the index set $L^{(1)}$ which corresponds to the minimum of F(L). If $L^{(1)} \neq L^{(0)}$ repeat step 2) using this new index set. The second step should be repeated only a fixed number of times before proceeding to step 4). If $L^{(1)} = L^{(0)}$, then proceed to step 4).
4) Assuming that the final error index set from 3) is given by L, compute the t transmission error estimates.
5) Reduce L by following the ANSD Reduction Rule. If the index set is reduced, re-compute the error amplitudes and repeat the reduction rule.
6) Once the final error index has been found, compute the estimate of the information vector.

In the next chapter, these thresholds are used to test the effectiveness of the ANSD algorithm.

CHAPTER VII

SIMULATION RESULTS

Contained in this chapter are the results of five numerical simulations. The purpose of these simulations are threefold. First, the source coding properties of real number (RN) BCH and Reed-Solomon (RS) codes need to be verified. Recall, the main source coding property describes the mean squared error given that the error locations are known ($M1_{RN}$), versus the number of errors. An additional source coding property described weighted codes. Simulated results are needed to verify these properties.

The second purpose is to verify the channel coding properties using fixed error positions and fixed syndrome magnitudes. Recall, for a given error location set and mean syndrome magnitude, the probability of a decoding failure, P_{DF}, was bounded in Chapter V. Simulated results are needed to test the accuracy of these bounds. In addition, for multiple error correction codes, several decoding methods are available: Prony's method, the nearest subspace decoding (NSD) method, and finally, the approximate nearest subspace decoding method (ANSD). The true performance of these decoding methods compared to the theoretical bounds needs to be illustrated.

The third purpose is to explore the joint source-channel coding performance of RN BCH and RS codes. Since the code rate of non-trivial, single error correcting RN codes is fairly high (i.e. $K/N = (N-2)/N \cong 1$), the source coding advantages of these codes is not very significant. The decision to use a single error correcting RN code will be based upon whether the channel coding performance is acceptable for a given signal to noise ratio. For this reason, the joint source-channel coding performance of single error cor-

recting RN codes is not simulated.

It is far more interesting and illustrative to simulate the performance of multiple error correcting RN codes. For these codes, the code rates are usually lower, and thus, the potential source coding advantages are greater. In addition, unlike the single error correcting codes, the average channel coding performance is not equal to the worst case performance.

Recall, in Chapter V, a minimum angle was derived which bounded the worst case channel coding performance for multiple error correcting codes. In addition, an average angle which estimated the average channel coding performance as a function of the SNR and the mean syndrome magnitude was derived; however, this angle could not be analytically justified. Even if this average angle is accurate, obtaining the average channel coding performance would still require a guess at an average syndrome magnitude. But, since the relationship of the syndrome magnitude to the actual error amplitudes for multiple error correcting codes is not as straight forward as it is for single error correcting codes (one would expect that an average syndrome magnitude would be a function of the number of errors), the average channel coding performance for multiple error correcting codes is probably best examined by simulation.

Again, five simulations were performed. The first two simulations investigate the source coding properties of real number error correction codes, while the third and fourth investigate the channel coding properties. The last simulation examines the average channel coding performance and also the joint source-channel coding properties. By using the results derived in chapters V and VI, the theoretical performance of RN error correction codes can be compared with the numerical results of these five simulations.

Source Coding Simulations

Two source coding simulations were performed. The first simulation computes the mean squared error obtained with a four error correcting (15,7) BCH code for two sets of fixed error positions with two different types of noise. The second simulation gives an example of a (7,3) weighted code.

Simulation #1

After constructing a (15,7) BCH code, identical, independent Gaussian samples were generated. These samples formed the length 7 information words, and they were subsequently transformed using the real generator matrix. Noise was added to the codewords in two different ways:

(1) White Gaussian noise was generated at a SNR of 30 dB. This was added to the codewords.

(2) The codewords were quantized to 6 bits with a uniform quantizer optimized for Gaussian data. For this case, the SNR was also approximately 30 dB.

For a given number of errors ranging from 0 to 4 with fixed positions, (it was assumed that the error positions were correctly decoded), the mean squared error ($M1_{RN}$) of the data estimate was computed. Using 500 iterations, $M1_{RN}$ was recorded and averaged for a variety of error positions.

The error positions can be divided into two sets. The first set consisted of error positions that were interleaved in the fashion described in Chapter V. The second set used consecutive error positions. For reference, the interleaved positions were given by {0}, {0,2}, {0,2,4}, {0,2,4,6}; while {0}, {0,1}, {0,1,2}, {0,1,2,3} were the consecutive positions.

Recall, that the interleaved positions corresponded to case resulting in the highest probability of a decoding failure using a NSD decoder. These error positions are the worst case for the channel coding analysis.

The consecutive error positions turn out to be the worst case for the source coding analysis. Recall that $M1_{RN}$ was a function of the error positions. Consecutive error positions result in the highest level of $M1_{RN}$.

Figures VII.1 and VII.2 show $M1_{RN}$ relative to $M1_{FF} = \sigma_q^2$ in decibels for the interleaved and consecutive error positions, respectively. In the figures, the solid line represents the theoretical value, computed by using Theorem V.7. The experimental values which correspond to the Gaussian noise are represented by an "o", while an "x" depicts the experimental values resulting from the uniform quantizer noise. The dashed line corresponds to the theoretical average value of $M1_{RN}$, which was previously plotted in Figure V.9.

Figure VII.1 shows that the experimental agrees quite well with the theoretical even in the case where the noise is truly quantization noise. It is seen that the interleaved case results in better than average source coding performance. For the interleaved error positions, the real number code always outperforms the finite field code since all values are negative. (The 0 dB line represents the finite field code performance.)

Figure VII.2 shows results which also agree with the theory. In this case however, the source coding performance is poor. Even at only 3 errors, $M1_{RN}$ for the consecutive error positions is higher than the mean squared error for the finite field code. Again, on the average, the source coding performance of the real number code is superior to the finite field code. However, it must be emphasized that perfect decoding has been assumed.

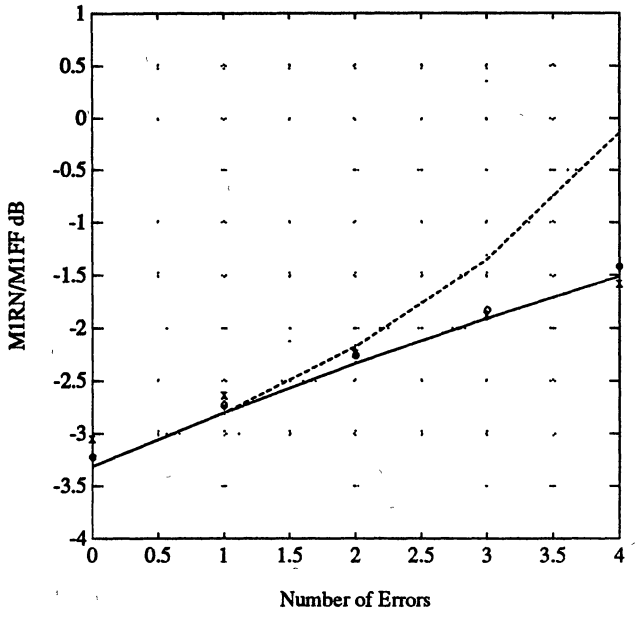


Figure VII.1: M1 vs. Number of Errors For Interleaved Positions

solid = theory; dashed = average; o = Gaussian Noise; x = Quantizer Noise.

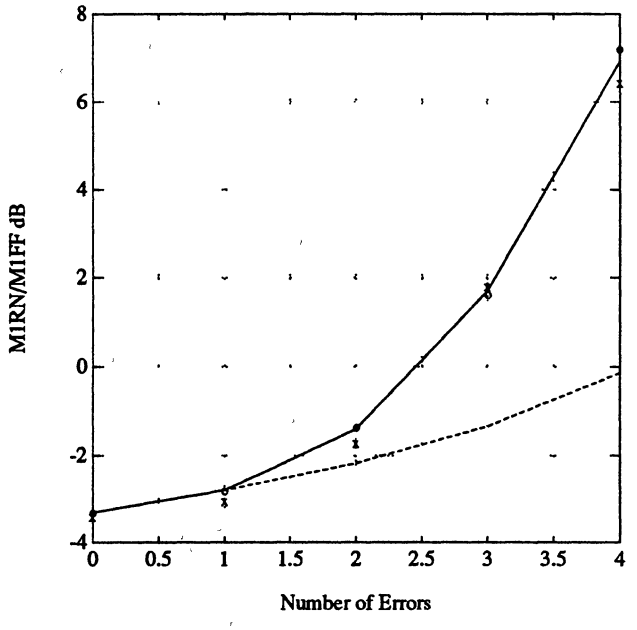


Figure VII.2: M1 vs. Number of Errors For Consecutive Positions

solid = theory; dashed = average; o = Gaussian Noise; x = Quantizer Noise.

Simulation #2

The second simulation investigates the source coding properties of a (7,3) BCH weighted code. A weighted generator matrix was constructed by creating a 3 by 3 weighting matrix as described by Theorem V.8. The weighting matrix is given by

$$W = \begin{bmatrix} 0.775 & 0 & 0 \\ 0 & 1.095 & 0 \\ 0 & 0 & 1.095 \end{bmatrix}.$$

If the information word is given by $d = [d_1, d_2, d_3]$, then the corresponding weights as specified by (V.26) are given by

$$\alpha_1 = \frac{5}{3}, \quad \alpha_2 = \alpha_3 = \frac{5}{6}.$$

Note that

$$\sum_{i=1}^{3} \frac{1}{\alpha_i} = 3 = K,$$

as Theorem V.8 states. The conjugacy constraint for the BCH code forces the second and third weights to be equal. Assuming that there are no errors, since

$$\frac{5/3}{5/6} = 2,$$

the estimate of the first data element of the information word should have an average MSE that is 3 dB higher than the MSE of the second and third elements.

Using 500 iterations and the two different noise sources described in the first simulation, the average mean squared error relative to the noise variance was computed and tabulated. With no transmission errors and no weighting, the MSE of each element of the information word should be 3/7 or -3.68 dB lower than what would be expected with an equivalent finite field code.

With the weighted code, the theoretical MSE of d_1 should be -1.46 dB while the theoretical MSE for d_2 and d_3 are -4.46 dB. Experimentally, the results agree fairly close to the theoretical values and are given as follows:

MSE in dB	d_1	d_2	d_3
Theoretical	-1.46	-4.46	-4.46
Gaussian Noise	-1.75	-4.76	-4.74
Quantizer Noise	-0.94	-4.25	-3.54

To reiterate, the weighted generator matrix allows certain data elements to be represented more precisely than other data elements. In this example, the second and third elements are assumed to be more important than the first since they have smaller values of α. The channel coding properties of this code remain unchanged by the weighting procedure as shown by Theorem V.8.

If transmission errors are present, then the MSE will depend upon the error positions. The theoretical values of the MSE can be found by explicitly calculating the covariance matrix of $\hat{d}$. Again, it must be assumed that the error positions have been correctly decoded.

Channel Coding Simulations

The presented source coding results, both theoretical and simulated, assumed that the error positions were always known or had been perfectly determined. However, this is not the case.

In Chapter V, it was emphasized that there is a non-zero probability of a decoding failure for real number codes. For single error correcting codes, (V.40) gave an upper bound for this probability as a function of the signal to noise ratio and the magnitude of

the syndrome vector for a given code. For multiple error correcting codes, (V.43a) generalized (V.40) and gave an upper bound for the probability of a decoding failure as a function of the SNR and the mean syndrome magnitude. Using the estimated average angle, (V.43b) gives an estimate for the average probability of a decoding failure.

In this section, the results of two channel coding simulations are presented. The first simulation investigates the probability of a decoding failure for two single error correcting codes. The second simulation looks at the probability of a decoding failure for the (15,7) BCH code.

Simulation #1

Figures V.5 and V.6 show the theoretical probability of a decoding error versus the signal to noise ratio for the (7,5) and (15,13), single error correcting BCH codes. P_{DF} was plotted by using (V.40) with seven different syndrome magnitudes: .125, .25, .5, .75, 1.0, 1.25, 1.5. Recall, that in the single error correcting case, the syndrome magnitude is equal to the magnitude of the transmission error. If an average transmission error magnitude is calculated from a specified channel model and quantizer, then the theoretical average channel performance can be found.

Using randomly generated Gaussian noise, the first channel coding simulation numerically computed the probability of a decoding failure for a number of specified SNR values using the same error magnitudes. A total of 1000 iterations were used.

Figures VII.3 and VII.4 show these results along with the previously presented theoretical curves. Note that since only 1000 iterations were performed for each error magnitude at each SNR, the smallest non-zero probability of a decoding failure that could be calculated experimentally was $1/1000 = 10^{-3}$. In all the cases where the real number code did not fail, the experimental "probability" was zero and is not shown on the semi-log plot.

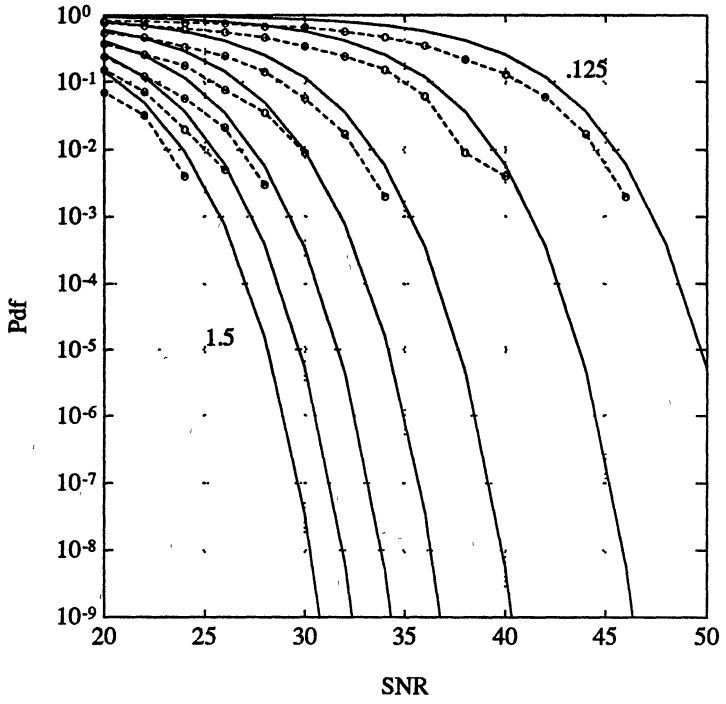


Figure VII.3: P_{DF} vs. SNR for (7,5) BCH Code

solid - theory; dashed & "o' - experimental

The experimental results show that Equation (V.40) appears to be a good upper bound for the probability of a decoding failure for single error correcting codes. Also, since the theoretical and experimental results closely agree, the previous conclusion that high signal to noise ratios will be required to reliably guard against moderately sized transmission errors is still valid.

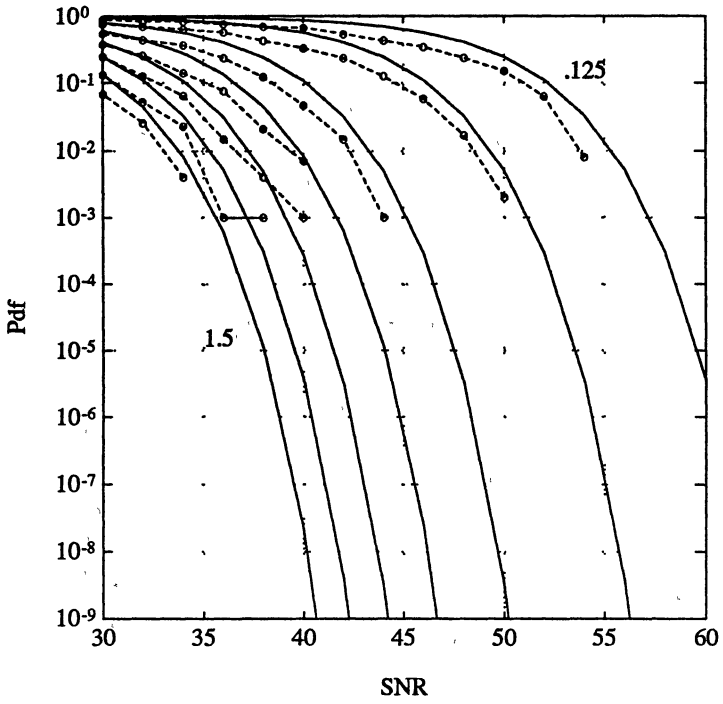


Figure VII.4: P_{DF} vs. SNR for (15,13) BCH Code

solid - theory; dashed & "o' - experimental

Finally note that the values for SNR vary from Figure VII.3 to Figure VII.4. The larger blocklength (15,13) code requires a higher signal to noise ratio, since the minimum angle between the syndrome error subspaces decreases. Again, this agrees with the theory.

Simulation #2

The second channel coding simulation tests the channel coding performance for a (15,7) BCH code. Recall, in Chapter V, the worst case minimum angles as a function of the number of errors were presented. This case occurred when the error locations were interleaved. The interleaved error index set used in this simulation are the same as the interleaved positions in the first source coding simulation.

For each error index set L, with $|L| = 0, 1, \ldots, 4$, a transmission error vector was calculated such that the resulting syndrome is in the direction of the principal vector that corresponds to $\theta_{min}(L)$. In other words, for a given L, the error vector corresponds to a syndrome that lies in the worst possible direction.

Of course, for the interleaved error index sets, L is the worst of all possible index sets. Thus, the performance for the interleaved error locations represents the worst possible channel coding performance for the (15,7) code. The minimum angles for this code are given in Chapter V as $\theta_{min}(1)$, $\theta_{min}(2)$, $\theta_{min}(3)$, and $\theta_{min}(4)$.

Since the channel coding performance depends upon the both the magnitude of the mean syndrome vector and the level of the quantization noise, these quantities had to be fixed. For a given magnitude of the mean syndrome vector and a given SNR, a total of 100 trials were performed for each of the four error index sets.

A trial consisted of generating a noisy syndrome and then estimating the error location set by three methods: (1) Prony's method, (2) Approximate NSD method, and (3) NSD method. (The ANSD method computed a maximum of four discrete neighborhoods to perform the local search.) For each of these methods, the estimated error location set was compared against the true error location set and a tally of all the decoding failures was recorded.

Thus, for a given SNR and syndrome magnitude, the percent number of failures could be plotted against the number of errors. Next, the syndrome magnitude was changed and the procedure repeated. A total of four magnitudes were tested: .125, .5, 1.0, and 1.5. Finally, the signal to noise ratio of the quantization noise was changed. The three values of SNR were 36 dB, 48 dB, and 60 dB.

By using (V.43a), the theoretical probability of a decoding failure as a function of the number of errors can be approximated. By multiplying P_{DF} by 100, one can get the total percentage of failures which would be theoretically predicted. These values can be compared with the simulated results.

In Figure VII.5, four sets of results for SNR = 36 dB. are presented: the theoretical results, the simulated results from using only Prony's method, the simulated results from using the ANSD method, and the simulated results from using the true NSD method. The results are plotted separately for each of the four syndrome magnitudes. Figures VII.6 and VII.7 show the results for SNR = 48 dB and SNR = 60 dB, respectively.

In all the cases, the theoretical results (solid) provide an upper bound on the percent number of failures when compared to the true NSD results (marked with an '*'). As expected, the true NSD method performs the best, Prony's method performs the worst, while the ANSD method is a compromise between the other two.

From these results, it is obvious that for these error vectors, the real number code is not very reliable. In many cases, it fails 100% of the time, especially when SNR = 36 dB. However, it should be emphasized that this is the worst possible case. On the average, the results should be better.

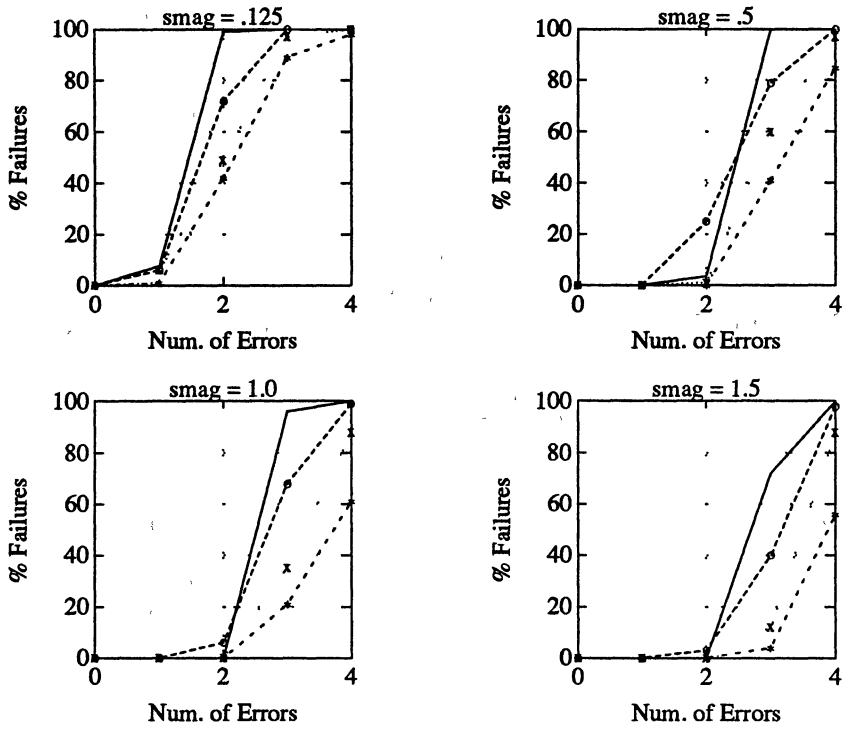


Figure VII.5: Percentage of Failures vs. Number of Errors for (15,7) BCH Code With Interleaved Error Locations and SNR = 36 dB.

solid - theory; 'o' - Prony; 'x' - ANSD; '*' - NSD

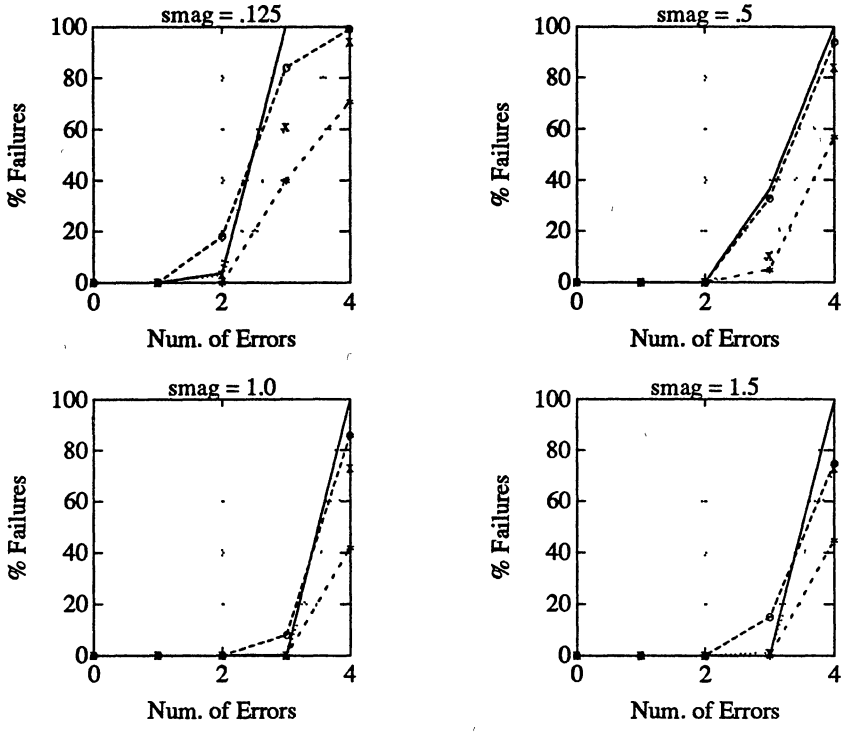


Figure VII.6: Percentage of Failures vs. Number of Errors for (15,7) BCH Code With Interleaved Error Locations and SNR = 48 dB.

solid - theory; 'o' - Prony; 'x' - ANSD; '*' - NSD

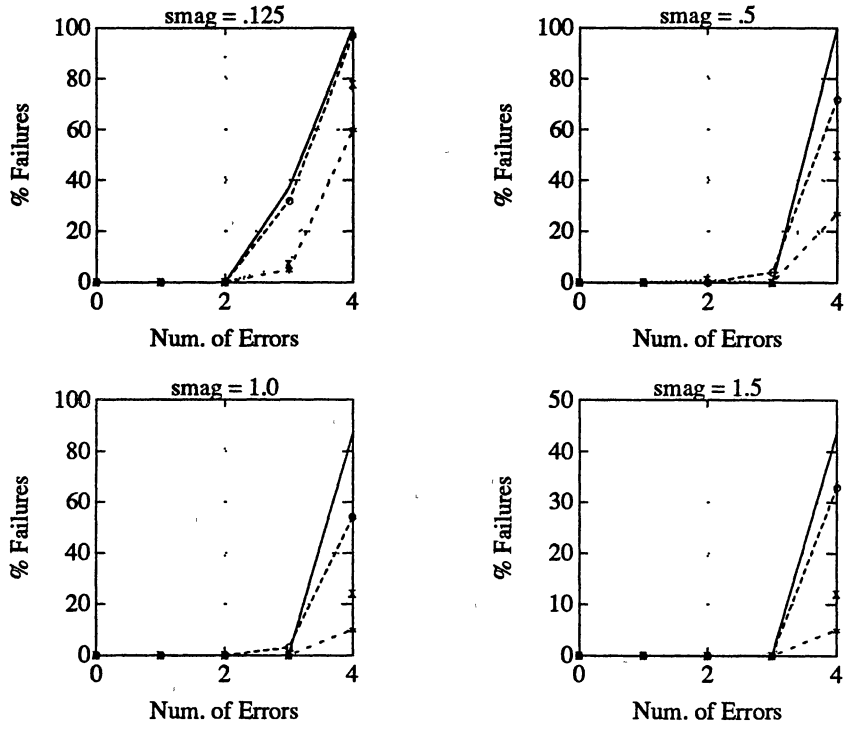


Figure VII.7: Percentage of Failures vs. Number of Errors for (15,7) BCH Code With Interleaved Error Locations and SNR = 60 dB.

solid - theory; 'o' - Prony; 'x' - ANSD; '*' - NSD

The results in Figures VII.5 - VII.7 used a fixed syndrome magnitude in order to compare the theoretical worst case results with the simulated worst case results. This simulation does not give any indication of the average performance. Clearly, it would not be realistic to fix the syndrome magnitude and get an average performance, since as the number of transmission errors increases, the syndrome magnitude is likely to increase. For this reason, the true average performance is best simulated by using a more realistic test scenario. This includes quantizing the codeword and generating discrete transmission errors caused by bit changes in randomly chosen codeword elements.

The final simulation in this chapter is such a test. In this simulation, not only the average channel coding performance, but also the source coding performance is examined.

Joint Source-Channel Coding Simulation

In this section, rather than isolating the source or the channel coding properties, the (15,7) code is tested in a more realistic environment. Again, using Gaussian data, a random information vector is generated and encoded. Then a uniform quantizer optimized for Gaussian data is used to quantize the codeword. A random number of errors, between 0 and 4, is generated. The locations are also chosen randomly. The actual transmission errors are generated by randomly altering the quantized codeword elements that correspond to the error locations. Finally, the ANSD algorithm was used to decode the received vector. Again, the maximum number of steps in the local search was limited to 4.

Using the quantized codeword, two cases were simulated. The first case used 6 bits, while the second used 8 bits. These cases roughly correspond to SNRs of 30 db and 41 db respectively. A total of 2500 iterations were run for each case. Since the number of errors was uniformly distributed between 0 and 4, out of the 2500 iterations, it would

be expected that 500 iterations would have no errors, 500 would have one error, etc. Hopefully, this will provide enough data to get some insight into the true performance of the (15,7) BCH code.

By keeping record of the true error locations and the estimated error locations, a tally of how many failures versus the number of errors can be created. In Figure VII.8, such a tally is plotted for the two SNRs. In order to conform with previous simulated and theoretical results, the number of failures is given as a percentage. By looking again at Figure V.7, in light of the simulated results for t=4 errors, either the average syndrome magnitude is small (on the order of .25 for the 6 bit case and .1 for the 8 bit case) or the estimated average performance is optimistic. By examining the syndromes obtained in the simulation, it must be concluded that the estimated performance is very optimistic.

Since the mean of the syndrome is not limited to a fixed value, the results in Figure VII.8 are not compared to any theoretical results. Rather, these results are used to indicate the average performance of the (15,7) BCH code since a good theoretical average performance has not been obtained. As expected, the 8 bit case out performs the 6 bit case. In addition, the highest number of failures occurred when there were 4 errors. This is consistent with what the theory predicts.

By keeping record of the actual information word and the estimated information word, the source coding performance can be analyzed and compared to the theoretical performance. First, $M1_{RN}$ relative to $M1_{FF}$ can be plotted versus the number of errors. Recall, $M1_{RN}$ is the mean squared error for the real number code given that there is not a decoding error. $M1_{FF}$ corresponds to mean squared error with no decoding errors using a finite field code. This latter value is equal to the quantization noise variance.

To compute $M1_{RN}$, all the decoding failures were removed and the average mean squared error of the data estimate was calculated. The results for the 6 and 8 bit cases are

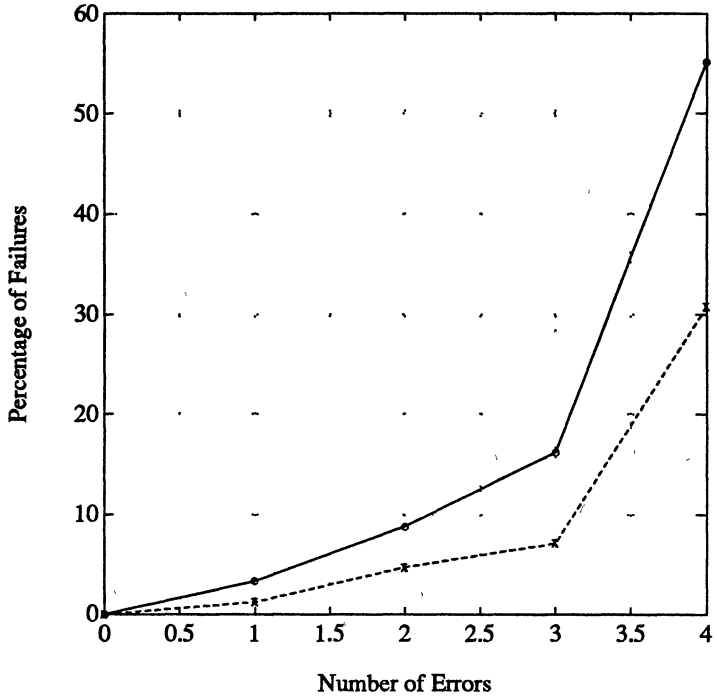


Figure VII.8: Percentage of Failures vs. Number of Errors for (15,7) BCH Code for Random Error Locations

'o' - 6 bits, 'x' - 8 bits

plotted in Figure VII.9. In addition, the theoretical average source coding results from Figure V.9 are reproduced for comparison. Recall, the theoretical results neglect the channel coding performance and are independent of the quantization noise level.

Both the 6 and 8 bit cases do better than the finite field results (the 0 dB line), however, only the 8 bit case follows the shape of the theoretical curve. Decoding the 6 bit case is much more difficult than decoding the 8 bit case, and despite the dependency of

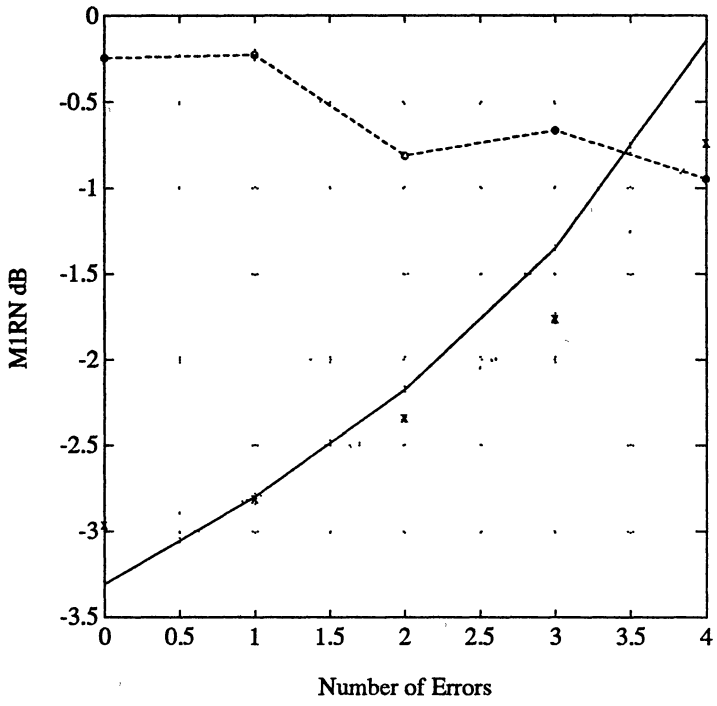


Figure VII.9: $M1_{RN}$ Relative to $M1_{FF}$ in dB vs Number of Errors

'o' - 6 bits, 'x' - 8 bits, solid - theory

the ANSD thresholds upon the noise variance, the ANSD algorithm consistently over-predicted the number of errors in the 6 bit case. This helps to explain the almost constant level of $M1_{RN}$.

Decoding the 8 bit case was much more straight forward and accurate. (As the results in Figure VII.8 depicted.) For this reason, the source coding results of the 8 bit case are closer to the theory.

Of course, the true joint source channel coding performance must include the mean squared error for the cases where the decoder failed. In Figure VII.10, the experimental total mean squared error is plotted. Unlike the results in Chapter V where the probability of a decoding failure was assumed to be negligible (this means that $M1_{RN} \approx MSE_{RN}$), for the experimental results, the effects of the decoding failures could not be ignored.

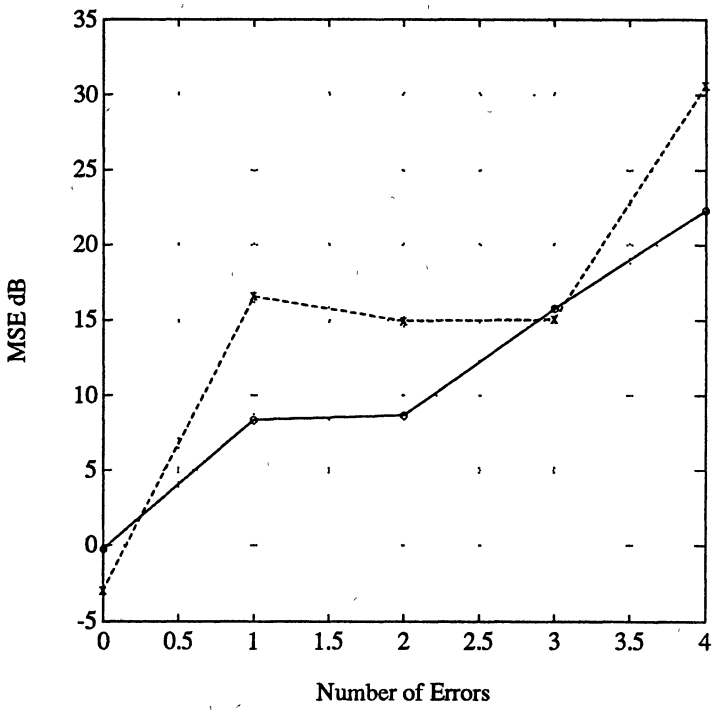


Figure VII.10: MSE_{RN} Relative to MSE_{FF} in dB vs Number of Errors

'o' - 6 bits, 'x' - 8 bits

The results in Figure VII.10 suggest that the total MSE for this real number code will almost never be better than the finite field alternative, since for any non-zero number of errors, MSE_{RN} is greater than MSE_{FF}.

Even though the joint source-channel coding performance, (as measured by the mean squared error), of real number BCH and RS codes seems to be inferior to the performance of a comparable finite field code, there still might be cases where a real number code could be preferable. If other distortion measures are considered, the impulsive distortion contributions corresponding to the decoding failures might not be as objectionable as they are with the quadratic distortion measure.

Besides the reliability concerns of real number codes at low and moderate SNRs, another disadvantage of real number codes is the fact that the ANSD algorithm requires a significant number of calculations. (If the SNR is very high, it is possible that Prony's method will be sufficient.) It was once thought that the natural implementation of real number codes on digital signal processors was a bonus, [Mar84], however, since many finite field decoders can be purchased as a single chip, [Ber87], it appears that finite field codes are also easier to implement.

CHAPTER VIII

SUMMARY AND CONCLUSIONS

The main objective of this investigation is to examine the joint source-channel coding properties of real number BCH and Reed-Solomon codes in the presence of additive noise. The investigative approach taken by the author was to first try to isolate the source coding properties from the channel coding effects by assuming perfect decoding. That is, it was assumed that the correct error locations were always determined by the channel decoder.

By assuming perfect decoding, the main source coding property relates the mean squared error of the estimated information word to the singular values of the deleted generator matrix. By averaging over all possible error locations, the source coding performance as a function of the number of errors was obtained. It was shown that average source coding performance for a (15,7) code was always superior to a comparable finite field code. However, for a (19,11) code, only when there were three or fewer errors was the average source coding performance superior. In general, for a fixed number of parity frequencies, as the blocklength increases, the source coding properties of real number codes will degrade.

Further source coding results generalized the basic normalization of the generator matrix and created a weighted code. Weighted codes allow certain data elements to be estimated more accurately than others. The construction of these codes is given in Chapter V (Theorem V.8). One idea that is emphasized is that the source coding properties depend only upon the generator matrix. On the other hand, the channel coding properties

solely depended upon the parity check matrix.

The second step in the investigative approach was to examine the channel coding properties by bounding the probability of a decoding failure. Starting with the easiest case, the channel coding properties were investigated by first examining single error correcting BCH codes. An optimal decoding rule was discussed, the nearest subspace decoding (NSD) rule, and bounds on the probability of a decoding failure were derived for this rule. It was found that the probability of a decoding failure is a function of the level of quantization noise, the magnitude of the transmission error, and the parameters of the code. In general, it was revealed that even for small codes, high signal to noise ratios are required to reliably correct moderately sized errors. This is a major drawback for real number codes, since it severely limits their possible applications.

The results for single error correcting codes were generalized to multiple error correction codes. It was shown that the worst case performance for multiple error correction codes can be tremendously poor. An estimate of average performance was not nearly so bad, however, this estimate turned out to be quite optimistic. The true average performance lies somewhere in between the worst case and this estimate. In addition, the average performance is difficult to verify by simulation since so many possible syndrome error subspaces exist.

Decoding multiple error correcting codes with Prony's method gave poor performance, while decoding with the NSD method proved to be prohibitively expensive. For this reason, an approximate NSD (ANSD) method was developed. It attempts to combine the accuracy of the NSD, but without all the expense. Simulation results show that the performance of the ANSD rule appears to be better than Prony's but not as good as a true NSD decoder. The ANSD rule still requires a considerable amount of computations.

A measure of the joint source-channel coding performance was obtained mostly by the final simulation of a four error correcting (15,7) code. A theoretical measure of the joint source-channel performance proved to be elusive since it requires a given quantizer and channel model. However, the final simulation shows that when looking at the overall mean squared error, it appears that a comparable finite field code will be superior. The assumption that the probability of a decoding failure is negligible does not hold. Even if P_{DF} is small, for those few times when a decoding failure is made, the MSE is very large compared to the finite field code performance. These impulses of error tend to increase the total MSE of a real number code significantly.

This is not to say that a finite field code will always be superior for all applications. It was remarked in Chapter VII, that for different distortion measures (other than the quadratic distortion measure), real number codes might be desirable. Such a measure could not give as much weight to occasional impulses of distortion.

In general, it is felt that for most applications where the source coding properties of the data are extremely important, (for instance, most low rate coding of real valued data), the real number code will be inferior to the more flexible and robust finite field techniques. For those high fidelity applications where bandwidth or storage capacity is abundant, then real number BCH and Reed-Solomon codes might prove to be useful.

Future Work and Possible Applications

A short-coming of this investigation is the lack of an accurate measure of the average channel coding performance for multiple error correcting codes. Part of the reason for this short-coming is that in order to compute the average channel coding of a given code, specific information about the channel and the quantizer are needed. Future work in real number error correction codes might attempt to analyze the average channel performance for specific applications.

Once a channel model and a quantizer are given, then a statistical distribution for the error amplitudes can be found. (This would be easiest if the quantizer is a uniform quantizer since the step sizes are all equal.) Given the distribution of the error amplitudes, it should be possible to find an average syndrome magnitude as a function of the number of errors. Given this information, the estimated average channel coding performance derived in Chapter V could be used. Of course, a computer simulation should be performed to verify the results.

If real number codes are proven to have some useful applications, then further work into the optimality of RN BCH and RS could be performed. For example, the single error correcting BCH codes are optimal in the sense that their error syndrome subspaces are equally spaced. No such result exists for the multiple error correcting versions.

Some possible applications might include the transmission or storage of high fidelity music. Such an application might have the required signal to noise ratio. Other applications might include the storage or transmission of high contrast images. Of course, the transmission and storage of real numbered computer data seems natural provided that some sort of compression is required.

REFERENCES

[Beh88] Behrens, R. T. & Scharf, L. L., "Parameter Estimation in the Presence of Low Rank Noise", *IEEE Proceedings of Asilomar Conference on Circuits, Systems, and Computers*, Pacific Grove, CA, Nov. 1988, pp. 341-344.

[Ber68] Berlekamp, E. R., *Algebraic Coding Theory*, McGraw-Hill Inc., New York, 1968.

[Ber87] Berlekamp, E. R., Peile, R. E., & Pope, S. P., "The Application of Error Control to Communications", *IEEE Communications Magazine*, Vol. 25, No. 4, April 1987, pp. 44-57.

[Bha83] Bhargava, V. K., "Forward Error Correction Schemes for Digital Communications", *IEEE Communications Magazine*, January 1983, pp. 11-19.

[Bla79] Blahut, R. E., "Transform Techniques for Error Control Codes", *IBM Journal of Research and Development*, Vol. 23, No. 3, May 1979, pp. 299-315.

[Bla83] Blahut, R. E., *The Theory and Practice of Error Control Codes*, Addison-Wesley Publishing Co., Reading, MA, 1983.

[Bla85] Blahut, R. E., "Algebraic Fields, Signal Processing, and Error Control", *Proceedings of the IEEE*, Vol. 73, No. 5, May 1985, pp. 874-893.

[Bla87] Blahut, R. E., *Principles and Practice of Information Theory*, Addison-Wesley Publishing Co., Reading, MA, 1987.

[Bla90] Blahut, R. E., *Digital Transmission of Information*, Addison-Wesley Publishing Co., Reading, MA, 1990

[Bos60] Bose, R. C. & Ray-Chaudhuri, D. K., "On a Class of Error Correcting Binary Group Codes", *Information and Control*, Vol. 3, March 1960, pp. 68-79.

[Chi64] Chien, R. T., "Cyclic Decoding Procedures for Bose-Chaudhuri-Hocquenghem Codes", *IEEE Transactions on Information Theory*, Vol. IT-10, October 1964, pp. 357-363.

[Con80] Conte, S. D. & de Boor, C., *Elementary Numerical Analysis: An Algorithmic Approach*, McGraw-Hill Book Co., New York, 1980.

[Cur74] Curtis, C. W., *Linear Algebra: An Introductory Approach*, Springer-Verlag New York, 1974.

[Dep88] Deprette, E. F. (Editor), *SVD and Signal Processing: Algorithms, Applications, and Architectures*. Elsevier Science Publishers B. V. (North Holland), 1988.

[Dew88] Dewilde, P. & Deprette, E. F., "Singular Value Decomposition: An Introduction", in *SVD and Signal Processing: Algorithms, Applications, and Architectures*. E. F. Deprette (Editor), Elsevier Science Publishers B. V. (North-Holland), 1988.

[Gal68] Gallager, R. G., *Information Theory and Reliable Communication*, John Wiley and Sons Inc., New York, 1968.

[Gol83] Golub, G. H. & Van Loan, C. F., *Matrix Computations*, Johns Hopkins University Press, Baltimore, MA, 1983.

[Gra80] Gray, R. M., Buzo, A., Gray, A. H. Jr., & Matsuyama, Y., "Distortion Measures for Speech Processing", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, No. 4, August 1980, pp. 367-376.

[Ham50] Hamming, R. W., "Error Detecting and Error Correcting Codes", *Bell System Technical Journal*, Vol. 29, April 1950, pp.147-160.

[Hil56] Hildebrand, F. B., *Introduction to Numerical Analysis*, McGraw-Hill Inc. 1956.

[Hoc59] Hocquenghem, A., "Codes correcteur d'erreurs", *Chiffres*, Vol. 2, 1959, pp. 147-156.

[Hua88] Hua, Y. & Sarkar, T. K., "Perturbation Analysis of TK Method for Harmonic Retrieval Problems", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 36, No. 2, February 1988, pp. 228-240.

[Hua90] Hua, Y. & Sarkar, T. K., "Matrix Pencil Method for Estimating Parameters of Exponentially Damped/Undamped Sinusoids in Noise", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 38, No. 5, May 1990, pp. 814-824.

[Jai89] Jain, A. K., *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[Jay84] Jayant, N. S. & Noll, P., *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[Kay88] Kay, S. M., *Modern Spectral Estimation: Theory and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[Kle80] Klema, V. C. & Laub, A. J., "The Singular Value Decomposition and Some Applications", *IEEE Transactions on Automatic Control*, Vol. AC-25, No. 2, April 1980, pp. 164-176.

[Kum85] Kumaresan, R., "Rank Reduction Techniques and Burst Error Correction Decoding in Real/Complex Fields", *IEEE Proceedings of Asilomar Conference on Circuits, Systems, and Computers*, Pacific Grove, CA, Nov. 1985, pp. 457-461.

[Law74] Lawson, C. L. & Hanson, R. J., *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[Lid84] Lidl, R. & Pilz, G., *Applied Abstract Algebra*, Springer-Verlag New York Inc., 1984.

[Llo82] Lloyd, S. P., "Least Squares Quantization in PCM", *IEEE Transactions on Information Theory*, Vol. IT-28, No. 2, March 1982, pp. 129-137.

[Mac77] MacWilliams, F. J. & Sloane, N. J. A., *The Theory of Error-Correcting Codes*, North-Holland Publishing Co., 1977.

[Mar87] Marple, S. L. Jr., *Digital Spectral Analysis with Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[Mar81] Marshall, T. G. Jr., "Real Number Transform and Convolutional Codes", *Proceedings of the 24th Midwest Symposium on Circuits and Systems*, Albuquerque, NM, 1981, pp. 650-653.

[Mar82] Marshall, T. G. Jr., "Methods for Error Correction with Digital Signal Processors", *Proceedings of the 25th Midwest Symposium on Circuits and Systems*, Houghton, MI, August 1982, pp. 1-5.

[Mar83a] Marshall, T. G. Jr., "Decoding of Real-Number Error-Correcting Codes", *Proceedings of the IEEE Global Telecommunications Conference*, San Diego, CA, Nov. 1983, pp. 1299-1303.

[Mar83b] Marshall, T. G. Jr., "Review of Real-Number Error Correction Codes and Their Implementation with Digital Signal Processors", *Proceedings of the International Conference on Communications*, Boston, MA, Vol. 1, 1983, pp. 313-317.

[Mar84] Marshall, T. G. Jr., "Coding of Real-Number Sequences for Error Correction: A Digital Signal Processing Problem", *IEEE Journal on Selected Areas in Communications*, Vol. SAC-2, No. 2, March 1984, pp 381-392.

[Mar85] Marshall, T. G. Jr., "Codes for Error Correction Based Upon Interpolation of Real Number Sequences", *IEEE Proceedings of Asilomar Conference on Circuits, Systems, and Computers*, Pacific Grove, CA, Nov. 1985, pp. 202-206.

[Mar86] Marshall, T. G. Jr., "Signal Restoration Viewpoints for Estimation Errors in Discrete-Time Signals", *IEEE Proceedings of Asilomar Conference on Circuits, Systems, and Computers*, Pacific Grove, CA, Nov. 1986, pp. 562-566.

[Mar87] Marshall, T. G. Jr., "Removing Noise Pulses form Frequency Constrained Signals", *Proceedings of the IEEE International Conference on Communications*, June 1987, pp. 997-1000.

[Mar88] Marshall, T. G. Jr. & Kolczynski, R. J., "Trapping Burst Errors in Discrete-Time Signals", *IEEE Proceedings of Asilomar Conference on Circuits, Systems, and Computers*, Pacific Grove, CA, Nov. 1988, pp. 341-344.

[Max60] Max, J., "Quantizing for Minimum Distortion", *IRE Transactions on Information Theory*, Vol. IT-6, March 1960, pp. 7-12.

[Mpl87] Marple, S. L. Jr., *Digital Spectral Analysis with Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[Nai90] Nair, V. S. S. & Abraham, J. A., "Real-Number Codes for Fault-Tolerant Matrix Operations On Processor Arrays", *IEEE Transactions on Computers*, Vol. 39, No. 4, April 1990, pp. 426-435.

[Pet60] Peterson, W. W., "Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes", *IRE Transactions on Information Theory*, Vol. IT-6, September 1960, pp. 459-470.

[Pet72] Peterson, W. W. & Weldon, E. J. Jr., *Error-Correcting Codes*, The MIT Press, Cambridge, MA, 1972.

[Pro95] de Prony, Baron (Gaspard Riche), "Essai experimental et analytique, etc.", *L'ecole Polytechnique, Paris*, Vol. 1, No. 2, 1795, pp. 24-76.

[Rah87] Rahman, M. A. & Yu, K., "Total Least Squares Approach for Frequency Estimation Using Linear Prediction", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, No. 10, October 1987.

[Ree60] Reed, I. S. & Solomon, G., "Polynomial Codes over Certain Finite Fields", *Journal of the Society of Industrial and Applied Mathematics*, Vol. 8, June 1960, pp. 300-304.

[Rob87] Roberts, R. A. & Mullis, C. T., *Digital Signal Processing*, Addison-Wesley, Reading, MA, 1987.

[Sch87] Scharf, L. L., Mathys, P., Behrens, R. T., "Rank Reduction for Decoding Linear Block Codes over the Complex Field", *IEEE Proceedings of Asilomar Conference on Circuits, Systems, and Computers*, Pacific Grove, CA, Nov. 1987, pp. 559-563.

[Sha48] Shannon, C. E., "A Mathematical Theory of Communication", *Bell System Technical Journal*, Vol. 27, July 1948, pp. 379-423.

[Shi77] Shilov, G. E., *Linear Algebra*, Dover Publications, New York, NY, 1977.

[Skl88] Sklar, B., *Digital Communications: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[Spr82] Sprague, D. L. & Marshall, T. G. Jr., "A Hadamard Transform Code for Error Correction over the Real Numbers", *Proceedings of the 25th Midwest Symposium on Circuits and Systems*, Houghton, MI, August 1982, pp. 9-13.

[Str88] Strang, G., *Linear Algebra and its Applications*, Harcourt Brace Jovanovich, Publishers, San Diego, CA, 1988.

[Tre82] Tremain, T. E., "The Government Standard Linear Prediction Coding Algorithm: LPC-10", *Speech Technology*, April 1982, pp. 40-49.

[Tuf82] Tufts, D. W., & Kumaresan, R., "Estimation of Frequencies of Multiple Sinusoids: Making Linear Prediction Perform Like Maximum Likelihood", *Proceedings of the IEEE*, Vol. 70, No. 9, September 1982, pp. 975-989.

[Wol67] Wolf, J. K., "Decoding of Bose-Chaudhuri-Hocquenghem Codes and Prony's Method for Curve Fitting", *IEEE Transactions on Information Theory*, October 1967, pg. 608.

[Wol83a] Wolf, J. K., "Analog Codes", *Proceedings of the International Conference on Communications*, Boston, MA, Vol. 1, 1983, pp. 310-312.

[Wol83b] Wolf, J. K., "Redundancy, the Discrete Fourier Transform, and Impulse Noise Cancellation", *IEEE Transactions on Communications*, Vol. COM-31, No. 3, March 1983, pp. 458-461.

APPENDIXES

APPENDIX A

MATRIX ALGEBRA: REVIEW AND NOTATION

The purpose of this appendix is to review some of the basic concepts and terminology of matrix algebra. Also, since notation can vary from text to text, a secondary purpose is to define the author's notation.

The term matrix algebra is used instead of linear algebra since linear transformations will always be viewed as a matrix. A basis is always assumed to be given. Also, several numerical properties relevant to computations with matrices are discussed. These ideas usually fall into the field of numerical analysis rather than linear algebra.

The material contained in this appendix has been drawn primarily from Golub & Van Loan, [Gol83], Curtis, [Cur74], and Strang, [Str88]. Two additional references were Shilov, [Shi77], and Conte & de Boor, [Con80]. By no means is this review intended to be complete; rather, it merely attempts to present the definitions and theorems which are pertinent to this report. The proofs for all the theorems can be found inside the given references.

A fundamental concept is that of a vector space over a field. The two fields of concern in this report are the real field, denoted by $\mathbf{R}$, and the complex field, $\mathbf{C}$. The definitions and theorems is this appendix will all be given using only the real field for convenience. Similar results are repeated in the complex field only when the notation is new or when the extension of a result from the real to the complex field is not obvious. A vector space over the real field can be defined as follows:

DEFINITION A.1: A *vector space X over* **R** (**R** is referred to as the scalar field) is a non-empty set X of objects, called *vectors*, together with the two operations: addition and scalar multiplication, such that

$$x_1 + x_2 \in X, \quad \forall \quad x_1, x_2 \in X$$

$$\alpha x \in X, \quad \forall \quad \alpha \in \mathbf{R}, x \in X.$$

Also, for $x, x_1, x_2 \in X$, $\alpha, \beta \in \mathbf{R}$, the operations are assumed to satisfy the following axioms:

(1) $x + (x_1 + x_2) = (x + x_1) + x_2$, and $x_1 + x_2 = x_2 + x_1$.

(2) $\exists$ vector $0 \in X$ such that $x + 0 = x \ \forall x \in X$.

(3) For each x, $\exists$ a vector $-x$ such that $x + (-x) = 0$.

(4) $\alpha(x_1 + x_2) = \alpha x_1 + \alpha x_2$

(5) $(\alpha + \beta)x = \alpha x + \beta x$.

(6) $(\alpha\beta)x = \alpha(\beta x)$.

(7) $1x = x$.

An example of a real vector space is the set of all real n-tuples denoted by $\mathbf{R}^n$. Most vectors of this type will be column vectors, i.e. if $x \in \mathbf{R}^n$, then

$$x = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix},$$

with $x_i \in \mathbf{R}$. If x is a row vector, it will be given by $x^T = [x_1, \ldots, x_n]$. Another example is the set of all $m \times n$ real matrices denoted by $\mathbf{R}^{m \times n}$. If $A \in \mathbf{R}^{m \times n}$, then

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}.$$

Frequently, the columns of A will be treated as vectors in $\mathbf{R}^m$. In this case, A will be written as

$$A = [a_1, \ldots, a_n].$$

Hopefully, there should not be any confusion between the columns of a matrix and scalar elements of a vector.

The following definitions are also fundamental:

DEFINITION A.2: A subset V of a vector space X is called a *subspace of* X if V is itself a vector space.

A set of vectors $\{x_1, \ldots, x_k\} \subset X$ is called *linearly dependent* if there exists scalars, $\alpha_1, \ldots, \alpha_k$ not all of which are zero, such that

$$0 = \alpha_1 x_1 + \cdots + \alpha_k x_k.$$

A set of vectors which is not linearly dependent is called *linearly independent.*

DEFINITION A.3: Given a set of vectors $\{x_1, \ldots, x_k\} \subset X$, then *span*$(x_1, \ldots, x_k)$ is the set of all linear combinations of $\{x_1, \ldots, x_k\}$.

It can be shown that if $V = span(x_1, \ldots, x_k)$ then V is a subspace of X. The vector space X is said to be *spanned by* $\{x_1, \ldots x_k\}$, if every $x \in X$ can be written as a linear combination of $\{x_1, \ldots x_k\}$.

DEFINITION A.4: A linearly independent set $\{x_1, \ldots, x_m\}$ which spans a vector space X is said to be a *basis* for X.

In the above case, the *dimension of* X is m or $dim(X) = m$. This report is only concerned with finite dimensional vector spaces. For example, $dim(\mathbf{R}^m) = m$ and $dim(\mathbf{R}^{m \times n}) = mn$.

In the context of this report, a matrix is generally viewed as a linear map or transformation from one vector space to another, rather than as a vector. The notation $A : X \rightarrow Y$ denotes that the matrix A maps a vector space X into a vector space Y.

Suppose $B \in \mathbf{R}^{n \times n}$. Then B is *nonsingular* or *invertible* if there exists a matrix $B^{-1} \in \mathbf{R}^{n \times n}$ such that

$$BB^{-1} = I_n,$$

where I_n is the $n \times n$ identity matrix. The following definition will lead to a condition for when a square matrix is invertible.

The *column (row) rank* of the matrix A is equal to the number of independent columns (rows) of A. It can be shown that the column rank of A equals the row rank of A. Thus, the column rank and row rank are shortened to just the rank of A or simply rank(A). This leads to the following result:

THEOREM A.5: The matrix $B \in \mathbf{R}^{n \times n}$ is invertible iff rank(B) = n.

Now suppose $A: X \rightarrow Y$. Then associated with A are two important subspaces.

DEFINITION A.6: The set $Im(A) = \{y \in Y \mid y = Ax \text{ for some } x \in X\}$ is called the *image of A*. (Sometimes *Im(A)* is called the range or rowspace of A and is denoted by R(A).)

THEOREM A.7: *Im(A)* is a subspace of Y.

DEFINITION A.8: The set $Ker(A) = \{x \in X \mid Ax = 0\}$ is called the *kernel of A*. (Sometimes *Ker(A)* is called the null space of A and is denoted by N(A).)

THEOREM A.9: *Ker(A)* is a subspace of X.

The dimensions of these two subspaces are related by the following important theorem:

THEOREM A.10: Let $A: X \rightarrow Y$, and rank(A) = r $\leq$ dim(X). Then

$$\begin{aligned} dim(X) &= dim(Im(A)) + dim(Ker(A)) \\ &= r + dim(Ker(A)). \end{aligned} \tag{A.1}$$

When $r = dim(X)$, then A is said to be of *full rank*. If A is square and of full rank, then A^{-1} exists.

Besides the inverse of A, there is also the transpose of A. If $A \in \mathbf{R}^{m \times n}$, $A = \{a_{ij}\}, i = 1,\ldots,m, j = 1,\ldots,n$, then the *transpose* of A is given by $A^T = \{a_{ji}\}$. A is called *symmetric* if $A = A^T$.

In the complex field, conjugation usually coincides with transposition. This is denoted by $A^H = \{a_{ji}^*\}$ where * denotes complex conjugation. A complex matrix A is called *Hermitian* if $A = A^H$.

Symmetric and Hermitian are two special classes of matrices. Another matrix class of interest is the Vandermonde matrices.

A matrix $V \in \mathbf{R}^{(n+1)\times(n+1)}$ of the form

$$V = V(x_0, x_1, \ldots, x_n) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_n \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ x_0^n & x_1^n & \cdots & x_n^n \end{bmatrix}$$

is called a *Vandermonde* matrix. It can be shown that $V(x_0, x_1, \ldots, x_n)$ is nonsingular if $x_0, x_1, \ldots, x_n$ are all distinct.

Two vectors $x_1, x_2 \in \mathbf{R}^n$ are said to be *orthogonal* if $x_1^T x_2 = 0$. A set of vectors $\{x_1, \ldots, x_k\}$ are said to be *orthonormal* if $x_i^T x_j = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$. In addition, an orthogonal matrix can be defined. A matrix $B \in \mathbf{R}^{n \times n}$ is said to be *orthogonal* if $BB^T = B^T B = I_n$.

Thus, if B is orthogonal, then B is invertible and $B^{-1} = B^T$. For complex matrices, a matrix $B \in \mathbf{C}^{n \times n}$ is said to be *unitary* if $BB^H = B^H B = I_n$.

Two subspaces can also be orthogonal. If $X, Y \subset \mathbf{R}^n$ are subspaces and every $x \in X$ is orthogonal to every $y \in Y$, then X is said to be orthogonal to Y. An orthogonal complement of a given subspace is defined below:

DEFINITION A.11: If $X \subset \mathbf{R}^n$ is a subspace, then the *orthogonal complement* of X is given by[1]

$$X^{\angle} = \{x' \in \mathbf{R}^n \mid x^T x' = 0\}, \quad \forall x \in X.$$

THEOREM A.12: The orthogonal complement of X is a subspace.

THEOREM A.13: If $X \subset \mathbf{R}^n$ is a subspace, then

$$dim(X) + dim(X^{\angle}) = dim(\mathbf{R}^n). \tag{A.2}$$

THEOREM A.14: If $v \in \mathbf{R}^n, X \subset \mathbf{R}^n$ a subspace, then v can be represented uniquely as $v = x + x'$ for some $x \in X$ and $x' \in X^{\angle}$.

An orthogonal projection can also be defined. The idea of a projection from a vector space onto a subspace is fundamental.

---

1 The author must apologize for the inconvenient symbol, $\angle$. Strangely, a perpendicular sign was not readily available to the equation typesetter of the author's word processing program.

DEFINITION A.15: Let $X \subset \mathbf{R}^n$ be a subspace. $P \in \mathbf{R}^{n\times n}$ is the unique *orthogonal projection onto* X if

(1) $Im(P) = X$.

(2) $P^2 = P$.

(3) $P^T = P$.

The relationship between the orthogonal projection onto a subspace, and the orthogonal projection onto the subspace's complement is given in the following theorem:

THEOREM A.16: Let $P \in \mathbf{R}^{n\times n}$ be an orthogonal projection onto X, then $(I_n - P)$ is an orthogonal projection onto $X^{\perp}$.

The expression $x_1^T x_2$ defines an *inner product* between two vectors in $\mathbf{R}^n$. (An *outer product* will be denoted by $x_1 x_2^T \in \mathbf{R}^{n\times n}$.) The inner product can be used to define a vector norm. The 2-norm, or sometimes called the Euclidean norm, of a vector x is given by

$$\|x\|_2 = (x_1^2 + \cdots + x_n^2)^{1/2} = (x^T x)^{1/2}. \qquad (A.3)$$

Note that the 2-norm is invariant under multiplication by an orthogonal matrix, i.e. if $B^T B = I_n$, then $\|Bx\|_2^2 = x^T B^T B x = x^T x = \|x\|_2^2$. Norms provide a measure of distance between two vectors in a vector space.

Matrix norms also exist. Two matrix norms are of interest: the Frobenius norm and the 2-norm. They are defined as follows:

Let $A \in \mathbf{R}^{m\times n}$, then the *Frobenius norm* of A is given by

$$\|A\|_F = \left[\sum_{i=1}^{m} \sum_{j=1}^{n} (a_{ij})^2 \right]^{1/2}. \qquad (A.4)$$

Let $A \in \mathbf{R}^{m \times n}$, then the *2-norm* of A is given by

$$\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}. \qquad (A.5)$$

Note that the Frobenius norm is merely the Euclidean norm of the length *mn* column vector formed by stacking the n columns of A.

Now suppose $A \in \mathbf{R}^{m \times n}$, $m > n$, and rank(A) = n. Then the system of equations:

$$y = Ax, \quad x \in \mathbf{R}^n, y \in \mathbf{R}^m$$

is called *overdetermined.* In this case, no inverse of A exists, but a "pseudo-inverse", or sometimes called the L_2 *inverse*, can be found.

DEFINITION A.17: Given a matrix $A \in \mathbf{R}^{m \times n}$, with $m > n$, the unique matrix $A^+ \in \mathbf{R}^{n \times m}$ that satisfies

$$\min_{A' \in \mathbf{R}^{n \times m}} \|AA' - I_m\|_F$$

is called the *pseudo-inverse* of A.

The vector $x_{LS} \in \mathbf{R}^n$ that satisfies

$$\min_{x \in \mathbf{R}^n} \|y - Ax\|_2$$

is called the *least squares* solution to the above overdetermined system of equations. If A is of full rank, then it can be shown that the solution, x_{LS}, is given by

$$x_{LS} = A^+ y = (A^T A)^{-1} A^T y. \qquad (A.6)$$

Occasionally, there is a need to refer to the condition of a matrix. The condition of a matrix provides insight into the sensitivity of an overdetermined system of equations. Formally, the *condition of the matrix A* is given by

$$\kappa_2(A) = \|A\|_2 \|A^+\|_2. \qquad (A.7)$$

The condition is norm dependent, however, only the 2-norm will be used in this report so the norm subscript is usually dropped.

Roughly speaking, the condition of A is inversely proportionally to the relative distance A is from being rank deficient. A large condition number indicates that A is nearly rank deficient. In such a case, A is said to be *ill-conditioned*, and x_{LS} is very sensitive to perturbations in both the matrix A, and the vector y.

If the condition of A is one, then A is said to be *perfectly conditioned.* In general, when solving a system of overdetermined equations, a perfectly conditioned matrix is desirable. Condition numbers are easily computed with the Singular Value Decomposition described in Appendix B.

APPENDIX B

THE SINGULAR VALUE DECOMPOSITION

The purpose of this appendix is to present the Singular Value Decomposition (SVD) Theorem and some resulting useful properties. The majority of this material has been drawn from Golub & Van Loan, [Gol83], Klema & Laub, [Kle80], Lawson & Hanson [Law74], Strang, [Str88], and Dewilde & Deprettere, [Dew88]. The proofs for all the theorems presented here can be found in these references, so they are not repeated here. Only real matrices are discussed; the extension of these theorems and properties for complex matrices can be found in the references.

The SVD Theorem is one of the most important decomposition theorems in computational matrix algebra. It is intimately connected with the subjects of norms, pseudo-inverses, and condition numbers presented in Appendix A.

THEOREM B.1: (The Singular Value Decomposition Theorem) Let $A \in \mathbf{R}^{m\times n}$ with rank(A) = r. Then there exist orthogonal matrices $U \in \mathbf{R}^{m\times m}$ and $V \in \mathbf{R}^{n\times n}$ such that,

$$A = U\Sigma V^T, \tag{B.1}$$

where,

$$\Sigma = \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} \in \mathbf{R}^{m\times n},$$

$$U = [u_1, \ldots, u_m] \qquad u_i \in \mathbf{R}^m,$$

$$V = [v_1, \ldots, v_n] \qquad u_i \in \mathbf{R}^n,$$

and $S = diag(\sigma_1, \ldots, \sigma_r)$, with

$$\sigma_1 \geq \ldots \geq \sigma_r > 0.$$

The i[th] diagonal element of the matrix S is called the *i^{th} singular value* of A. The singular values are equal to the positive square roots of the eigenvalues of the positive semi-definite matrix A^TA. The columns of U are called the *left singular vectors* of A; these are orthonormal eigenvectors of A^TA. The columns of V are called the *right singular vectors* of A; these are orthonormal eigenvectors of AA^T.

It should be noted that the singular vectors are not in general unique. If a particular singular value is distinct, then the corresponding singular vectors are also unique.

Both the Frobenius and the 2-norm of A can be expressed in terms of the singular values of A. Specifically,

$$\|A\|_F^2 = \sigma_1^2 + \cdots + \sigma_r^2, \tag{B.2}$$

$$\|A\|_2^2 = \sigma_1^2. \tag{B.3}$$

The pseudo-inverse of A, denoted by A^+, can also be expressed by using the SVD. Let A be given as in Theorem B.1, then

$$A^+ = V\Sigma^+U^T, \tag{B.4}$$

where

$$\Sigma^+ = \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbf{R}^{n \times m},$$

and $S^{-1} = diag(1/\sigma_1, \ldots, 1/\sigma_r)$. The condition of A, is given by $\kappa(A) = \sigma_1/\sigma_r$.

The matrix A can be expanded into a sum of outer products using the SVD:

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T. \tag{B.5}$$

This leads to a very useful result for signal processing called the Matrix Approximation Theorem.

THEOREM B.2: (Matrix Approximation Theorem) Let $A \in \mathbf{R}^{m\times n}$ with rank(A) = r, and let k be an integer less than r, then

$$\min_{B\in\mathbf{R}^{m\times n},\ rank(B)=k} \|A-B\|_2 = \left\| A - \sum_{i=1}^{k} \sigma_i u_i v_i^T \right\|_2 = \sigma_{k+1}. \qquad (B.6)$$

It should be noted that the matrix B which achieves the above minimum is not unique for the 2-norm.

The subspaces $Im(A) \subset \mathbf{R}^m$ and $Ker(A) \subset \mathbf{R}^n$ were discussed in Appendix A. Orthonormal bases for both these subspaces and their orthogonal complements can be found using the SVD.

THEOREM B.3: Let $A:X \to Y$, $dim(X) = m$, $dim(Y) = n$, rank(A) = r, and

$$A = U\Sigma V^T = [U_1, U_2]\begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix}\begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix},$$

with $U_1 \in \mathbf{R}^{m\times r}$, $V_1 \in \mathbf{R}^{n\times r}$, $U_2 \in \mathbf{R}^{m\times m-r}$, $V_2 \in \mathbf{R}^{n\times n-r}$, then

(1) The columns of U_1 provide an orthonormal basis for $Im(A)$.

(2) The columns of U_2 provide an orthonormal basis for $Im(A)^{\perp} = Ker(A^T)$.

(3) The columns of V_1 provide an orthonormal basis for $Ker(A)^{\perp} = Im(A^T)$.

(4) The columns of V_2 provide an orthonormal basis for $Ker(A)$.

Finally, the following orthogonal projections onto the above subspaces can also be given in terms of the SVD of A:

THEOREM B.4: Let A be given as in Theorem B.3, then

(1) $P_{Im(A)}: Y \to Im(A); \quad P_{Im(A)} = U_1 U_1^T = AA^+.$

(2) $P_{Ker(A)^\perp}: X \to Ker(A)^\perp; \quad P_{Ker(A)^\perp} = V_1 V_1^T = A^+ A.$

(3) $P_{Im(A)^\perp}: Y \to Im(A)^\perp; \quad P_{Im(A)^\perp} = U_2 U_2^T = I - AA^+.$

(4) $P_{Ker(A)}: X \to Ker(A); \quad P_{Ker(A)} = V_2 V_2^T = I - A^+ A.$

A final useful theorem, concerns the resulting changes of the singular values of a matrix A, when a row has been deleted. This theorem belongs to a class of perturbation theorems for the singular value decomposition.

THEOREM B.5: Let $A \in \mathbf{R}^{m \times n}$, with $m > n$. Let $B \in \mathbf{R}^{(m-1) \times n}$ be formed by deleting a row from A. Then the ordered singular values β_i of B interlace the ordered singular values σ_i of A as follows:

$$\sigma_1 \geq \beta_1 \geq \sigma_2 \geq \beta_2 \geq \cdots \geq \sigma_n \geq \beta_n \geq 0.$$

2

VITA

John David Endsley

Candidate for the Degree of

Doctor of Philosophy

Thesis: JOINT SOURCE-CHANNEL CODING WITH REAL NUMBER BCH AND REED-SOLOMON CODES: THEIR PROPERTIES AND PERFORMANCE IN THE PRESENCE OF ADDITIVE NOISE

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Las Vegas, Nevada, September 20, 1963, the son of David and Joan Endsley.

Education: Graduated from Sandia High School, Albuquerque, New Mexico, in May 1981; received Bachelor of Science degree in Electrical Engineering from New Mexico State University in May, 1985; received Master of Science degree from the University of Colorado at Boulder in May 1988; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in May, 1991.

Professional Experience: Technical Staff, TRW Antenna Lab, Redondo Beach, California, June, 1985, to July 1986. Teaching Assistant, Department of Electrical Engineering, University of Colorado at Boulder, August, 1986, to May, 1987. Technical Staff, BDM Corporation, Boulder, Colorado, June, 1987, to July, 1988. Research Assistant, Department of Electrical Engineering, Oklahoma State University, August, 1988, to May, 1991.